

9.1

*IBM MQ Konfigurationsreferenz*

**IBM**

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 289 gelesen werden.

Diese Ausgabe bezieht sich auf Version 9 Release 1 von IBM® MQ und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuauflage geändert wird.

Wenn Sie Informationen an IBM senden, erteilen Sie IBM ein nicht ausschließliches Recht, die Informationen in beliebiger Weise zu verwenden oder zu verteilen, ohne dass eine Verpflichtung für Sie entsteht.

© **Copyright International Business Machines Corporation 2007, 2024.**

---

# Inhaltsverzeichnis

<b>Konfigurationsreferenz.....</b>	<b>5</b>
IBM MQ-Beispielkonfiguration für alle Plattformen.....	5
Verwendung der Kommunikationsbeispiele.....	7
Unterstützung mehrerer Threads — Pipelining.....	8
IBM MQ-Beispielkonfiguration für AIX.....	9
IBM MQ-Beispielkonfiguration für IBM i.....	15
IBM MQ-Beispielkonfiguration für Linux.....	31
IBM MQ-Beispielkonfiguration für Solaris.....	37
IBM MQ-Beispielkonfiguration für Windows.....	43
IBM MQ-Beispielkonfiguration für z/OS.....	49
IBM MQ-Beispielkonfiguration für z/OS unter Verwendung von QSGs.....	54
MQ-Beispielkonfiguration für z/OS unter Verwendung der gruppeninternen Warteschlangen- steuerung.....	62
Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen.....	70
IBM MQ-Dateiberechtigungen in /opt/mqm mit setuid für mqm.....	74
IBM MQ-Dateisystemberechtigungen unter Windows.....	76
Benennungseinschränkungen für Warteschlangen.....	77
Benennungseinschränkungen für andere Objekte.....	79
Auflösung des Warteschlangennamens.....	81
Was ist die Auflösung von Warteschlangennamen?.....	82
Wie Zielobjektattribute für Aliasnamen, ferne Warteschlangen und Clusterwarteschlangen aufgelöst werden.....	83
System-und Standardobjekte.....	84
Objekte der Windows-Standardkonfiguration.....	89
SYSTEM.BASE.TOPIC.....	90
Informationen zu Zeilengruppen.....	91
Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen.....	94
Kanalattribute.....	95
Kanalattribute und Kanaltypen.....	96
Kanalattribute in alphabetischer Reihenfolge.....	100
IBM MQ-Clusterbefehle.....	136
Befehle für Warteschlangenmanagerdefinitionen.....	137
Befehle für die Kanaldefinition.....	138
Befehle für die Warteschlangendefinition.....	141
DISPLAY CLUSQMgr.....	144
Befehle SUSPEND QMgr und RESUME QMgr im Cluster.....	146
REFRESH CLUSTER.....	147
RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen.....	148
Lastausgleich in Clustern.....	150
Asynchrones Verhalten von Clusterbefehlen unter z/OS.....	159
Kanalprogramme.....	160
Jobs für übergreifende Kommunikation unter IBM i.....	160
Kanalzustände unter IBM i.....	161
Beispiel für Nachrichtenkanalplanung für UNIX, Linux, and Windows.....	161
Erläuterung der Beispiele für UNIX, Linux, and Windows.....	161
Beispiel für UNIX, Linux, and Windows ausführen.....	165
Beispiel für Nachrichtenkanalplanung für IBM i.....	165
Erläuterung der Beispiele für IBM i.....	166
Beispiel für IBM i ausführen.....	170
Beispiel für IBM i erweitern.....	170
Beispiel für Nachrichtenkanalplanung für z/OS.....	170
Erläuterung der Beispiele für z/OS.....	171

Beispiel für z/OS ausführen.....	174
Beispiel für z/OS erweitern.....	175
Nachrichtenkanalplanungsbeispiel für z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange.....	175
Erläuterung des Beispiels der Gruppe mit gemeinsamer Warteschlange für z/OS.....	175
Definitionen der Gruppe mit gemeinsamer Warteschlange.....	177
Warteschlangenmanager QM3 - Beispiel für z/OS.....	178
Beispiel mit Gruppe mit gemeinsamer Warteschlange für z/OS ausführen.....	178
Einsatz eines Alias zum Verweis auf eine MQ-Bibliothek.....	179
Managed File Transfer -Konfigurationsreferenz.....	179
Verwendung von Umgebungsvariablen in MFT-Eigenschaften.....	179
Die MFT-Datei 'installation.properties'.....	180
Die MFT-Datei 'coordination.properties'.....	185
Die MFT-Datei 'command.properties'.....	191
Die MFT agent.properties-Datei.....	196
Die MFT-Datei 'logger.properties'.....	221
Java-Systemeigenschaften für MFT.....	232
SSL-Eigenschaften für MFT.....	233
SHA-2-CipherSpecs und -CipherSuites für MFT.....	240
Konfigurationsdateien der MFT-Dateiprotokollfunktion.....	240
Bibliothek SBFGCMD5.....	247
Thema 'SYSTEM.FTE'.....	249
Einstellungen von MFT-Agentenwarteschlangen.....	250
MFT-Systemwarteschlangen und der Systemabschnitt.....	252
Konventionen zum Benennen von MFT-Objekten.....	253
Statusnachrichten von MFT-Agenten.....	255
IBM MQ Internet Pass-Thru -Konfigurationsreferenz.....	256
Zusammenfassung der MQIPT-Eigenschaften.....	256
Globale MQIPT-Eigenschaften.....	262
MQIPT-Routeneigenschaften.....	264
mqzOSConnectService -Element.....	283
HTTP-Header, die mit dem MQ Service Provider verwendet werden können.....	287
<b>Bemerkungen.....</b>	<b>289</b>
Informationen zu Programmierschnittstellen.....	290
Marken.....	291


# Konfigurationsreferenz

Verwenden Sie die Referenzinformationen in diesem Abschnitt zur Unterstützung beim Konfigurieren von IBM MQ.

Die Konfigurationsreferenzinformationen werden in folgenden Unterabschnitten bereitgestellt:

## Zugehörige Tasks

Konfiguration

 z/OS konfigurieren

## IBM MQ-Beispielkonfiguration für alle Plattformen

In den Konfigurationsbeispielen werden die Aufgaben beschrieben, die zum Aufbau eines funktionierenden IBM MQ-Netztes auszuführen sind. Zu den Aufgaben gehört die Einrichtung von Sender- und Empfängerkanälen von IBM MQ für einen bidirektionalen Nachrichtenfluss zwischen den Plattformen über alle unterstützten Protokolle.

Wenn Sie andere Kanaltypen als Sender-Empfänger-Kanäle verwenden möchten, lesen Sie die Beschreibung des Befehls `DEFINE CHANNEL`.

Abbildung 1 auf Seite 5 ist eine konzeptuelle Darstellung eines einzelnen Kanals und der ihm zugeordneten IBM MQ-Objekte.

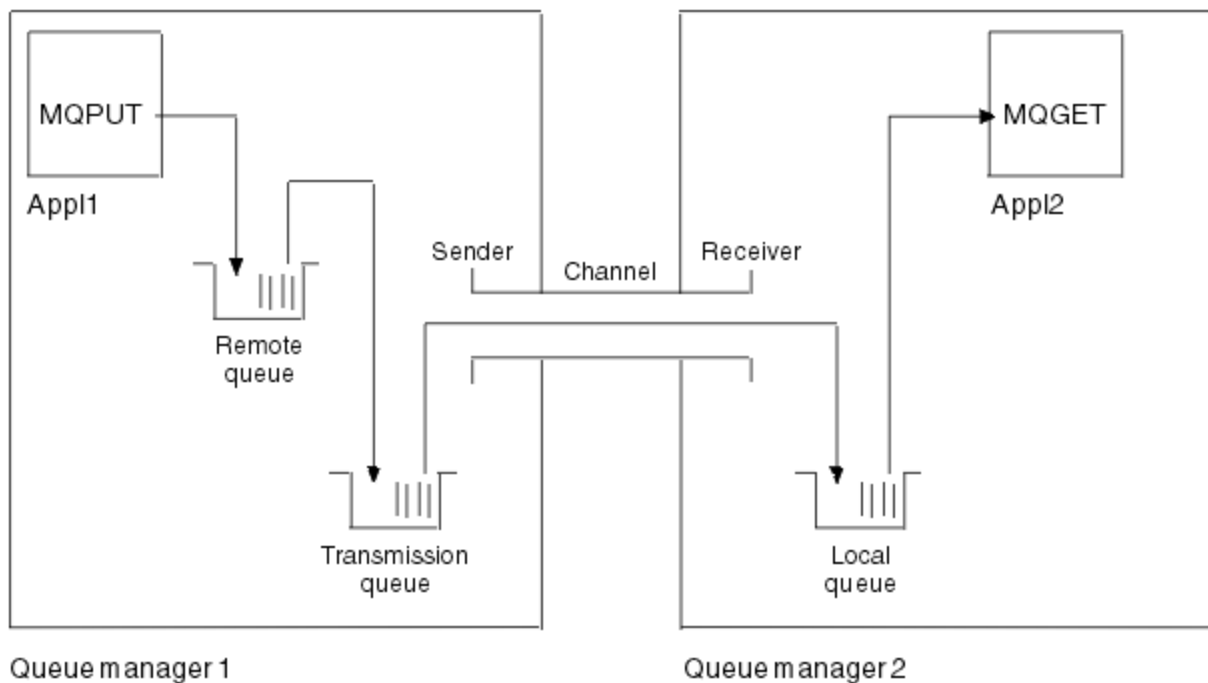


Abbildung 1. IBM MQ-Kanal, der in der Beispielkonfiguration einzurichten ist

Dies ist ein einfaches Beispiel, das eine Einführung in die grundlegenden Elemente des IBM MQ-Netztes geben soll. Es beinhaltet nicht die Auslösefunktion, die im Abschnitt [Kanäle auslösen](#) beschrieben wird.

Objekte in diesem Netz:

- eine ferne Warteschlange
- eine Übertragungswarteschlange
- Eine lokale Warteschlange
- ein Senderkanal

- fein Empfängerkanal

Appl1 und Appl2 sind Anwendungsprogramme; Appl1 reiht Nachrichten ein und Appl2 empfängt sie.

Appl1 reiht Nachrichten in eine ferne Warteschlange ein. In der Definition für diese ferne Warteschlange sind der Name eines Ziel-Warteschlangenmanagers, eine lokale Warteschlange auf diesem Warteschlangenmanager und eine Übertragungswarteschlange auf dem lokalen Warteschlangenmanager angegeben.

Wenn der Warteschlangenmanager die Anforderung von Appl1 zum Einreihen einer Nachricht in die ferne Warteschlange empfängt, ermittelt er anhand der Warteschlangendefinition, dass es sich um eine ferne Zieladresse handelt. Er stellt die Nachricht deshalb zusammen mit einem Übertragungsheader direkt in die Übertragungswarteschlange, die in der Definition angegeben ist. Die Nachricht bleibt in der Übertragungswarteschlange, bis der Kanal verfügbar wird; dies kann sofort der Fall sein.

Ein Senderkanal enthält in seiner Definition einen Verweis auf eine einzige Übertragungswarteschlange. Wenn ein Kanal gestartet wird, und ansonsten während seines normalen Betriebs, schaut er in dieser Übertragungswarteschlange nach und sendet eventuell darin enthaltene Nachrichten an das Zielsystem. Die Nachricht enthält in ihrem Übertragungsheader Informationen zur Zielwarteschlange und zum Warteschlangenmanager.



Die Beispiele für übergreifende Kommunikation beschreiben ausführlich die Erstellung jedes einzelnen der zuvor beschriebenen Objekte für verschiedene Plattformkombinationen.

Auf dem Ziel-Warteschlangenmanager sind Definitionen für die lokale Warteschlange und die Empfängerseite des Kanals erforderlich. Diese Objekte sind voneinander unabhängig und können deshalb in beliebiger Reihenfolge erstellt werden.

Auf dem lokalen Warteschlangenmanager sind Definitionen für die ferne Warteschlange, die Übertragungswarteschlange und die Senderseite des Kanals erforderlich. Da sowohl die Definition der fernen Warteschlange als auch die Kanaldefinition auf den Namen der Übertragungswarteschlange verweisen, ist es ratsam, die Übertragungswarteschlange zuerst zu erstellen.

## Netzinfrastruktur im Beispiel

Die Konfigurationsbeispiele setzen voraus, dass bestimmte Netzinfrastrukturen für bestimmte Plattformen vorhanden sind:




-  z/OS kommuniziert über einen 3745-Netzcontroller (oder funktional entsprechenden Controller), der an einen Token-Ring angeschlossen ist.
-  Solaris ist in einem benachbarten lokalen Netz (LAN) ebenfalls an einen 3745-Netzcontroller (oder funktional entsprechenden Controller) angeschlossen.
- Alle anderen Plattformen sind mit einem Token-Ring-Netz verbunden.

Außerdem wird vorausgesetzt, dass für SNA alle erforderlichen Definitionen im VTAM und im Netzsteuerprogramm (NCP) enthalten und aktiviert sind, sodass die über LAN angeschlossenen Plattformen über das Weitverkehrsnetz (WAN) kommunizieren können.

Entsprechend wird für TCP angenommen, dass die Namensserverfunktion verfügbar ist, entweder über einen Domänennamensserver oder über lokal vorhandene Tabellen (z. B. eine Hostdatei).

## Datenübertragungssoftware im Beispiel

Die Beispiele enthalten funktionsfähige Konfigurationen für folgende Netzsoftwareprodukte:

- SNA (Systems Network Architecture)
  -  IBM Personal Communications für Windows 5.9
  -  IBM Communications Server für AIX 6.3
  - Hewlett-Packard SNAplus2
  -  IBM i

- Data Connection SNAP-IX 7 oder höher
- OS/390 2.4
- TCP
  - **Windows** Microsoft Windows
  - **AIX** AIX 4.1.4
  - **Solaris** Sun Solaris 2.4 oder höher
  - **IBM i** IBM i
  - **z/OS** TCP für z/OS
- NetBIOS
- SPX

### Zugehörige Tasks

Verteilte Warteschlangensteuerung konfigurieren

Kommunikation mit anderen Warteschlangenmanagern unter z/OS einrichten

## Verwendung der Kommunikationsbeispiele

In den Beispielkonfigurationen werden die Tasks beschrieben, die auf einer einzelnen Plattform zum Konfigurieren der Kommunikation zu einer anderen Plattform ausgeführt werden. Dann werden die Task zum Einrichten eines funktionsfähigen Kanals für diese Plattform beschrieben.

Wo immer möglich, wird versucht, die Informationen so allgemein wie möglich zu halten. Deshalb müssen Sie sich nur auf die beiden relevanten Abschnitte beziehen, wenn Sie zwei beliebige Warteschlangenmanager auf verschiedenen Plattformen miteinander verbinden möchten. Alle Abweichungen oder Sonderfälle werden hervorgehoben. Sie können auch zwei Warteschlangenmanager miteinander verbinden, die auf derselben Plattform (auf verschiedenen Maschinen oder derselben Maschine) aktiv sind. In diesem Fall können alle Informationen dem einen zutreffenden Abschnitt entnommen werden.

**ULW** Bevor Sie beginnen, die Anweisungen für Ihre Plattform zu befolgen, müssen Sie unter UNIX, Linux®, and Windows verschiedene Umgebungsvariablen festlegen. Geben Sie dazu einen der folgenden Befehle ein:

- **Windows** Unter Windows:

```
MQ_INSTALLATION_PATH/bin/setmqenv
```

Dabei ist `MQ_INSTALLATION_PATH` das Verzeichnis, in dem IBM MQ installiert ist.

- **Linux** **UNIX** Auf UNIX and Linux-Systemen:

```
. MQ_INSTALLATION_PATH/bin/setmqenv
```

Dabei ist `MQ_INSTALLATION_PATH` das Verzeichnis, in dem IBM MQ installiert ist. Dieser Befehl legt die Umgebungsvariablen für die Shell fest, in der Sie zurzeit arbeiten. Wenn Sie eine andere Shell öffnen, müssen Sie erneut den Befehl eingeben.

Es gibt Beispiele, in denen Sie die in den Beispielkonfigurationen verwendeten Parameter finden. Zu jedem Parameter gibt es eine Kurzbeschreibung und Angaben dazu, wo die entsprechenden Werte in Ihrem System zu finden sind. Wenn Sie über eine eigene Gruppe von Werten verfügen, stellen Sie sicher, dass Sie diese Werte beim Durcharbeiten der Beispiele in diesem Abschnitt verwenden.

Die Beispiele decken nur Umgebungen ab, in denen kein Clustering verwendet wird. Informationen zur Konfiguration der Datenübertragung bei Verwendung des Clusterings finden Sie im Abschnitt Warte-

schlangenmanagercluster konfigurieren. Die hier angegebenen Werte für die Konfiguration der Kommunikation gelten weiterhin.

Es sind Beispielkonfigurationen für die folgenden Plattformen vorhanden:

- ▶ **AIX** „IBM MQ-Beispielkonfiguration für AIX“ auf Seite 9
- ▶ **IBM i** „IBM MQ-Beispielkonfiguration für IBM i“ auf Seite 15
- ▶ **Linux** „IBM MQ-Beispielkonfiguration für Linux“ auf Seite 31
- ▶ **Solaris** „IBM MQ-Beispielkonfiguration für Solaris“ auf Seite 37
- ▶ **Windows** „IBM MQ-Beispielkonfiguration für Windows“ auf Seite 43
- ▶ **z/OS** „IBM MQ-Beispielkonfiguration für z/OS“ auf Seite 49
- ▶ **z/OS** „IBM MQ-Beispielkonfiguration für z/OS unter Verwendung von QSGs“ auf Seite 54
- ▶ **z/OS** „MQ-Beispielkonfiguration für z/OS unter Verwendung der gruppeninternen Warteschlangensteuerung“ auf Seite 62

## IT-Zuständigkeiten

Die folgenden Erläuterungen sollen als Ausgangspunkt zum besseren Verständnis der in den Beispielen verwendeten Terminologie dienen:

- **Systemadministrator:** Die Person (oder Gruppe von Personen), die die Software für eine bestimmte Plattform installiert und konfiguriert.
- **Netzadministrator:** Die Person, die die LAN-Konnektivität, die LAN-Adresszuordnungen, die Namenskonventionen im Netz und andere mit dem Netz verbundenen Tasks steuert. Diese Person kann Mitglied der Systemverwaltungsgruppe oder einer separaten Gruppe sein.

In den meisten z/OS-Installationen ist eine Gruppe für die Aktualisierung der ACF/VTAM-, ACF/NCP- und TCP/IP-Software zur Unterstützung der Konfiguration zuständig. Die Mitglieder dieser Gruppe bilden die Hauptquelle für Informationen, die zur Herstellung einer Verbindung zwischen einer IBM MQ-Plattform und IBM MQ for z/OS benötigt werden. Außerdem können sie Netznamenskonventionen in LANs beeinflussen oder vorgeben. Deshalb müssen Sie deren Zuständigkeitsbereich überprüfen, bevor Sie eigene Definitionen erstellen.

- Ein bestimmter Typ von Administrator, zum Beispiel der CICS-Administrator, wird in den Fällen angegeben, in denen der Verantwortungsbereich der Person genauer beschrieben werden kann.

In den Abschnitten mit den Beispielkonfigurationen wird nicht versucht anzugeben, wer für jeden einzelnen Parameter zuständig ist und ihn festlegen darf. Im Allgemeinen sind mehrere Personen beteiligt.

### Zugehörige Konzepte

„IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5

In den Konfigurationsbeispielen werden die Aufgaben beschrieben, die zum Aufbau eines funktionierenden IBM MQ-Netzes auszuführen sind. Zu den Aufgaben gehört die Einrichtung von Sender- und Empfängerkanälen von IBM MQ für einen bidirektionalen Nachrichtenfluss zwischen den Plattformen über alle unterstützten Protokolle.

### Zugehörige Verweise

[setmqenv](#)

## Multi

## Unterstützung mehrerer Threads – Pipelining

Sie haben die Option, einem Nachrichtenkanalagenten (MCA) die Übertragung von Nachrichten über mehrere Threads zu gestatten. Dieser Prozess, als *Pipelining* bezeichnet, ermöglicht dem MCA, Nachrichten effizienter mit weniger Wartestatus zu übertragen, wodurch sich die Kanalleistung verbessern lässt. Für jeden MCA können maximal zwei Threads ausgeführt werden.



Pipelining wird über den Parameter *PipeLineLength* in der Datei "qm.ini" gesteuert. Dieser Parameter wird der Zeilengruppe CHANNELS hinzugefügt:

**PipeLineLength= 1 | Anzahl**

Dieses Attribut gibt die maximale Anzahl der von einem Kanal verwendeten parallel ausgeführten Threads an. Der Standardwert ist 1. Alle Werte größer als 1 werden wie der Wert 2 behandelt.

**Anmerkung:** Pipelining ist nur für TCP/IP-Kanäle effektiv.

Bei der Verwendung von Pipelining muss in der Konfiguration der Warteschlangenmanager an beiden Kanalenden der Parameter *PipeLineLength* auf einen größeren Wert als 1 festgelegt sein.

## Überlegungen zu Kanalexits

Pipelining kann aus folgenden Gründen bei einigen Exitprogrammen zu Ausfällen führen:

- Exits wurden nicht seriell aufgerufen.
- Exits werden von unterschiedlichen Threads abwechselnd aufgerufen.

Überprüfen Sie die Struktur Ihres Exitprogramms, bevor Sie Pipelining nutzen:

- Exits müssen auf allen Ausführungsstufen wiedereintrittsfähig sein.
- Wenn Sie MQI-Aufrufe verwenden, achten Sie darauf, nicht die gleiche MQI-Kennung zu verwenden, wenn der Exit aus verschiedenen Threads aufgerufen wird.

Ziehen Sie einen Nachrichtenexit in Betracht, der eine Warteschlange öffnet und bei allen nachfolgenden Aufrufen des Exits dessen Kennung für MQPUT-Aufrufe angibt. Dies ist im Pipelining-Modus nicht möglich, da der Exit aus verschiedenen Threads aufgerufen wird. Um dies zu vermeiden, halten Sie eine Warteschlangenkennung für jeden Thread bereit, und überprüfen Sie bei jedem Aufruf des Exits die Thread-ID.

## AIX IBM MQ-Beispielkonfiguration für AIX

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for AIX- und IBM MQ-Produkten.

In den Beispielen werden die folgenden Plattformen behandelt:

- **Windows** Windows
- **Solaris** Solaris
- **Linux** Linux
- **IBM i** IBM i
- **z/OS** z/OS
- VSE/ESA

Unter „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## AIX Eine LU 6.2-Verbindung einrichten

Beschreibt die für eine LU 6.2-Verbindung erforderlichen Parameter.

Aktuelle Informationen zum Konfigurieren von SNA über TCP/IP finden Sie in der folgenden IBM-Online-dokumentation: [Communications Server for AIX Library](#).

## AIX Eine TCP-Verbindung einrichten

Das Empfangsprogramm muss unbedingt vor jeglichen Kanälen gestartet werden. Es ermöglicht empfangenden Kanälen das automatische Starten nach dem Erhalt einer Anforderung von einem Senderkanal für eingehende Nachrichten.

Der IBM MQ-Befehl, über den das IBM MQ for TCP-Empfangsprogramm gestartet wird, lautet:

```
runmqclsr -t tcp
```

Wenn Sie das mit UNIX bereitgestellte TCP/IP-Empfangsprogramm verwenden möchten, können Sie alternativ die folgenden Schritte ausführen:

1. Bearbeiten Sie die Datei `"/etc/services"`.

**Anmerkung:** Hierzu müssen Sie sich als Superuser oder Root-Benutzer anmelden. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. Bearbeiten Sie die Datei `"/etc/inetd.conf"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie abgebildet hinzu, und ersetzen Sie dabei `MQ_INSTALLATION_PATH` durch das übergeordnete Verzeichnis, in dem IBM MQ installiert ist:

```
MQSeries stream tcp nowait root MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name]
```

3. Geben Sie den Befehl `refresh -s inetd` ein.

**Anmerkung:** Sie müssen in der Gruppe `"mqm"` den Zusatz **root** einfügen. Die Primärgruppe muss auf `"mqm"` gesetzt sein. Solange sich `"mqm"` in dem Satz der Gruppen befindet, können Sie die Befehle verwenden. Wenn Sie nur Anwendungen ausführen, die den Warteschlangenmanager verwenden, benötigen Sie keine Gruppenberechtigung für `"mqm"`.

## Wie geht es weiter?

Die Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Wechseln Sie zu [„IBM MQ for AIX-Konfiguration“](#) auf Seite 10.

## IBM MQ for AIX-Konfiguration

Definieren von Kanälen zum Abschließen der Konfiguration.

### Anmerkung:

1. Bevor Sie mit dem Installationsprozess beginnen, stellen Sie sicher, dass der Benutzer und die Gruppe `mqm` erstellt und das Kennwort festgelegt wurden.
2. Wenn die Installation fehlschlägt, weil nicht genügend Speicherplatz im Dateisystem verfügbar ist, können Sie die Größe wie folgt mit dem Befehl `smit C snaerhöhen`. (Geben Sie `df` ein, um den Status des Dateisystems anzuzeigen. Daraufhin wird der volle logische Datenträger angezeigt).

```
-- Physical and Logical Storage  
-- File Systems  
-- Add / Change / Show / Delete File Systems  
-- Journalled File Systems  
-- Change/Show Characteristics of a Journalled File System
```

3. Starten Sie einen beliebigen Kanal über den Befehl:

```
runmqchl -c channel.name
```

4. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert, wobei `MQ_INSTALLATION_PATH` das übergeordnete Verzeichnis angibt, in dem IBM MQ installiert ist.
5. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.

6. Unter AIX können Sie einen Trace der IBM MQ-Komponenten durch Eingabe der Standardtracebefehle für IBM MQ oder mithilfe des AIX-Systemtrace starten. Weitere Informationen zu IBM MQ Trace und AIX -Systemtrace finden Sie unter [Trace verwenden](#) .
7. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

## Basiskonfiguration

1. Erstellen Sie den Warteschlangenmanager aus der AIX-Befehlszeile mithilfe des folgenden Befehls:

```
crtmqm -u dlqname -q aix
```

Dabei gilt:

### ***aix***

ist der Name des Warteschlangenmanagers

### **-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

### **-u *dlqname***

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

2. Starten Sie den Warteschlangenmanager aus der AIX-Befehlszeile mithilfe des folgenden Befehls:

```
strmqm aix
```

, wobei *aix* der Name ist, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

3. Starten Sie **runmqsc** aus der AIX-Befehlszeile und erstellen Sie die Warteschlange für unzustellbare Nachrichten durch Eingabe des folgenden Befehls:

```
def ql (dlqname)
```

, wobei *dlqname* der Name ist, den die Warteschlange für unzustellbare Nachrichten bei Erstellung des Warteschlangenmanagers erhalten hat.

## **Kanalkonfiguration für AIX**

Enthält Informationen zum Konfigurieren des Warteschlangenmanagers für einen bestimmten Kanal und eine bestimmte Plattform.

Im folgenden Abschnitt finden Sie ausführliche Informationen zu der Konfiguration des AIX-Warteschlangenmanagers für die Implementierung des in [„IBM MQ-Beispielkonfiguration für alle Plattformen“](#) auf Seite 5 beschriebenen Kanals.

Es ist jeweils der MQSC-Befehl angegeben. Entweder starten Sie **runmqsc** in einer AIX-Befehlszeile und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for AIX und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werten im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.

Tabelle 1. Konfigurationsbeispiele für IBM MQ for AIX







ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<b>Definition für lokalen Knoten</b>			
A	Name des Warteschlangenmanagers		<b>AIX</b>
B	Name der lokalen Warteschlange		<b>AIX.LOCALQ</b>
  <b>Verbindung zu IBM MQ for Windows</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Windows“ auf Seite 46 angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>WINNT</b>
G	Name des Senderkanals (SNA)		<b>AIX.WINNT.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>AIX.WINNT.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.AIX.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.AIX.TCP</b>
  <b>Verbindung zu IBM MQ for Solaris</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 39 angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>Solaris</b>
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>Solaris</b>
G	Name des Senderkanals (SNA)		<b>AIX.SOLARIS.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>AIX.SOLARIS.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.AIX.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.AIX.TCP</b>
  <b>Verbindung zu IBM MQ for Linux</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 34 angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>LINUX</b>
G	Name des Senderkanals (SNA)		<b>AIX.LINUX.SNA</b>

Tabelle 1. Konfigurationsbeispiele für IBM MQ for AIX (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
H	Name des Senderkanals (TCP/IP)		<b>AIX.LINUX.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.AIX.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.AIX.TCP</b>
<p><b>IBM i</b> <b>IBM i</b> <b>Verbindung zu IBM MQ for IBM i</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für IBM i“ auf Seite 27 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>AS400</b>
G	Name des Senderkanals (SNA)		<b>AIX.AS400.SNA</b>
H	Name des Senderkanals (TCP)		<b>AIX.AS400.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>AS400.AIX.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>AS400.AIXTCP-</b>
<p><b>z/OS</b> <b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für z/OS“ auf Seite 51 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>MVS</b>
G	Name des Senderkanals (SNA)		<b>AIX.MVS.SNA</b>
H	Name des Senderkanals (TCP)		<b>AIX.MVS.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>MVS.AIX.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>MVS.AIX.TCP</b>
<p><b>z/OS</b> <b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Beispiel für die Konfiguration gemeinsamer Kanäle“ auf Seite 59 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>QSG</b>
D	Name der fernen Warteschlange		<b>QSG.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>QSG.SHAREDQ</b>
F	Name der Übertragungswarteschlange		<b>QSG</b>
G	Name des Senderkanals (SNA)		<b>AIX.QSG.SNA</b>

Tabelle 1. Konfigurationsbeispiele für IBM MQ for AIX (Forts.)			
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
H	Name des Senderkanals (TCP)		<b>AIX.QSG.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>QSG.AIX.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>QSG.AIX.TCP</b>

### AIX Senderkanaldefinitionen unter Verwendung der SNA in IBM MQ for AIX

Beispielbefehle:

```
def q1 (WINNT) +                               F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                       D
  rname(WINNT.LOCALQ) +                       E
  rqmname(WINNT) +                             C
  xmitq(WINNT) +                               F
  replace

def chl (AIX.WINNT.SNA) chltype(sdr) +        G
  trptype(lu62) +
  conname('WINNTCPIC') +                     17
  xmitq(WINNT) +                               F
  replace
```

### AIX Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for AIX

Beispielbefehle:

```
def q1 (AIX.LOCALQ) replace                    B

def chl (WINNT.AIX.SNA) chltype(rcvr) +      I
  trptype(lu62) +
  replace
```

### AIX TPN-Konfiguration für IBM MQ for AIX

Alternative Möglichkeit, sicherzustellen, dass SNA-Empfängerkanäle ordnungsgemäß aktiviert werden, wenn ein Senderkanal einen Datenaustausch einleitet.

Bei der Konfiguration von AIX Communications Server wurde ein LU 6.2-TPN-Profil erstellt, das den vollständigen Pfad zu einem ausführbaren TP-Programm enthält. Im Beispiel trug die Datei den Namen "u/interop/AIX.crs6a". Sie können einen beliebigen Namen wählen, es kann jedoch sinnvoll sein, den Namen Ihres Warteschlangenmanagers darin aufzunehmen. Die ausführbare Datei muss Folgendes enthalten:

```
#!/bin/sh
MQ_INSTALLATION_PATH/bin/amqcrs6a -m aix
```

Dabei steht *aix* für den Namen des Warteschlangenmanagers (A) und *MQ\_INSTALLATION\_PATH* ist das übergeordnete Verzeichnis, in dem IBM MQ installiert ist. Nachdem diese Datei erstellt wurde, aktivieren Sie deren Ausführung durch den Befehl:

```
chmod 755 /u/interop/AIX.crs6a
```

Alternativ zum Erstellen einer ausführbaren Datei können Sie den Pfad im Fenster "LU 6.2 TPN-Profil hinzufügen (Add LU 6.2 TPN Profile)" mithilfe von Befehlszeilenparametern angeben.

Indem Sie durch eine der beschriebenen Vorgehensweisen einen Pfad angeben, stellen Sie sicher, dass SNA-Empfängerkanäle richtig aktiviert werden, wenn ein Senderkanal einen Datenaustausch einleitet.

### **AIX** *Senderkanaldefinitionen über TCP in IBM MQ for AIX*

Beispielbefehle:

```
def ql (WINNT) + usage(xmitq) + replace F
def qr (WINNT.REMOTEQ) + rname(WINNT.LOCALQ) + rqmname(WINNT) + xmitq(WINNT) + replace D
E
C
F
def chl (AIX.WINNT.TCP) chltype(sdr) + trptype(tcp) + conname(remote_tcpip_hostname) + xmitq(WINNT) + replace H
F
```

### **AIX** *Empfängerkanaldefinitionen unter Verwendung von TCP in IBM MQ for AIX*

Beispielbefehle:

```
def ql (AIX.LOCALQ) replace B
def chl (WINNT.AIX.TCP) chltype(rcvr) + trptype(tcp) + replace J
```

## **IBM i** **IBM MQ-Beispielkonfiguration für IBM i**

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for IBM i- und IBM MQ-Produkten auf anderen Plattformen.

Außerdem werden die folgenden weiteren Plattformen behandelt:

- **Windows** Windows
- **AIX** AIX
- **Solaris** Solaris
- **Linux** Linux
- **z/OS** z/OS oder MVS
- VSE/ESA

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## **IBM i** **Konfigurationsparameter für eine LU 6.2-Verbindung**

Im folgenden Arbeitsblatt sind alle Parameter aufgelistet, die zum Einrichten der Kommunikation von einem IBM i-System zu einer der anderen IBM MQ-Plattformen erforderlich sind. Das Arbeitsblatt enthält Beispiele der Parameter, die in einer Arbeitsumgebung getestet wurden, und ermöglicht Ihnen, Ihre eigenen Werte einzutragen.

Notieren Sie die Werte für diese Konfiguration in dem Arbeitsblatt dieses Abschnitts. Verwenden Sie das Arbeitsblatt mit dem Arbeitsblatt in dem Abschnitt für die Plattform, zu der Sie eine Verbindung herstellen.

Zahlen in der *Referenzspalte* zeigen an, dass der Wert dem Wert im entsprechenden Arbeitsblatt an anderer Stelle in diesem Abschnitt entsprechen muss. Die folgenden Beispiele in diesem Abschnitt beziehen sich auf die Werte in der ID-Spalte dieser Tabelle.

Die Einträge in der Spalte *Parametername* werden im Abschnitt „Begriffserklärungen“ auf Seite 18 erläutert.

<i>Tabelle 2. Arbeitsblatt zur Konfiguration von SNA auf einem IBM i-System</i>				
<b>ID</b>	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>	<b>Benutzerwert</b>
<b>Definition für lokalen Knoten</b>				
1	ID des lokalen Netzes		<b>NETID</b>	
2	Name des lokalen Steuerpunkts		<b>AS400PU</b>	
3	LU-Name		<b>AS400LU</b>	
4	LAN-Zieladresse		<b>10005A5962EF</b>	
5	Subsystembeschreibung		<b>QCMN</b>	
6	Leitungsbeschreibung		<b>TOKENRINGL</b>	
7	Ressourcenname		<b>LIN041</b>	
8	Name des lokalen Transaktionsprogramms		<b>MQSERIES</b>	
<b>Windows Verbindung mit einem Windows-System</b>				
9	Netz-ID	2	<b>NETID</b>	
10	Steuerpunktname	3	<b>WINNTCP</b>	
11	LU-Name	5	<b>WINNTLU</b>	
12	Controllerbeschreibung		<b>WINNTCP</b>	
13	Einheit		<b>WINNTLU</b>	
14	Nebeninformationen		<b>NTCPIC</b>	
15	Transaktionsprogramm	7	<b>MQSERIES</b>	
16	LAN-Adapteradresse	9	<b>08005AA5FAB9</b>	
17	Mode	17	<b>#INTER</b>	
<b>AIX Verbindung mit einem AIX-System</b>				
9	Netz-ID	1	<b>NETID</b>	
10	Steuerpunktname	2	<b>AIXPU</b>	
11	LU-Name	4	<b>AIX-LU</b>	
12	Controllerbeschreibung		<b>AIXPU</b>	
13	Einheit		<b>AIX-LU</b>	
14	Nebeninformationen		<b>AIXCPIC</b>	
15	Transaktionsprogramm	6	<b>MQSERIES</b>	
16	LAN-Adapteradresse	8	<b>123456789012</b>	
17	Mode	14	<b>#INTER</b>	



Tabelle 2. Arbeitsblatt zur Konfiguration von SNA auf einem IBM i-System (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Solaris</b> <i>Verbindung mit einem Solaris-System</i>				
9	Netz-ID	2	NETID	
10	Steuerpunktname	3	SOLARPU	
11	LU-Name	7	SOLARLU	
12	Controllerbeschreibung		SOLARPU	
13	Einheit		SOLARLU	
14	Nebeninformationen		SOLCPIC	
15	Transaktionsprogramm	8	MQSERIES	
16	LAN-Adapteradresse	5	08002071CC8A	
17	Mode	17	#INTER	
<b>Linux</b> <i>Verbindung mit einem Linux-System (x86-Plattform)</i>				
9	Netz-ID	4	NETID	
10	Steuerpunktname	2	LINUXPU	
11	LU-Name	5	LINUXLU	
12	Controllerbeschreibung		LINUXPU	
13	Einheit		LINUXLU	
14	Nebeninformationen		LXCPCIC	
15	Transaktionsprogramm	7	MQSERIES	
16	LAN-Adapteradresse	8	08005AC6DF33	
17	Mode	6	#INTER	
<b>z/OS</b> <i>Verbindung mit einem z/OS-System</i>				
9	Netz-ID	2	NETID	
10	Steuerpunktname	3	MVSPU	
11	LU-Name	4	MVSLU	
12	Controllerbeschreibung		MVSPU	
13	Einheit		MVSLU	
14	Nebeninformationen		MVSCPIC	
15	Transaktionsprogramm	7	MQSERIES	
16	LAN-Adapteradresse	8	400074511092	
17	Mode	6	#INTER	
<b>Verbindung mit einem VSE/ESA-System</b>				
9	Netz-ID	1	NETID	
10	Steuerpunktname	2	VSEPU	

Tabelle 2. Arbeitsblatt zur Konfiguration von SNA auf einem IBM i-System (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
11	LU-Name	3	VSELU	
12	Controllerbeschreibung		VSEPU	
13	Einheit		VSELU	
14	Nebeninformationen		VSECPIC	
15	Transaktionsprogramm	4	MQ01	MQ01
16	LAN-Adapteradresse	5	400074511092	
17	Mode		#INTER	

## IBM i Begriffserklärungen

Eine Erläuterung der im Arbeitsblatt zur Konfiguration verwendeten Begriffe.

### 1 2 3

Ausführliche Informationen zum Ermitteln der konfigurierten Werte finden Sie unter [„Netzattribute finden“](#) auf Seite 19.

### 4 LAN-Zieladresse

Die Hardwareadresse des Token-Ring-Adapters des IBM i-Systems. Sie können den Wert mithilfe des Befehls DSPLIND *Leitungsbeschreibung* (6) ermitteln.

### 5 Subsystembeschreibung

Dieser Parameter ist der Name eines IBM i-Subsystems, das aktiv ist, während der Warteschlangenmanager verwendet wird. Der Name QCMN wurde verwendet, da es sich um das Kommunikationssystem von IBM i handelt.

### 6 Leitungsbeschreibung

Wenn dieser Parameter angegeben wurde, wird er im Beschreibungsfeld des Ressourcennamens angezeigt. Ausführliche Informationen finden Sie in [„Den Wert des Ressourcennamens ermitteln“](#) auf Seite 19. Ist der Wert nicht angegeben, müssen Sie eine Leitungsbeschreibung erstellen.

### 7 Ressourcename

Ausführliche Informationen zum Ermitteln des konfigurierten Werts finden Sie unter [„Den Wert des Ressourcennamens ermitteln“](#) auf Seite 19.

### 8 Name des lokalen Transaktionsprogramms

IBM MQ-Anwendungen, die versuchen, Daten mit dieser Workstation auszutauschen, geben einen symbolischen Namen für das Programm an, das auf der Empfangsseite ausgeführt werden soll. Dieser Name ist in der Kanaldefinition auf der Senderseite definiert. Verwenden Sie der Einfachheit halber wenn möglich immer den Transaktionsprogrammnamen MQSERIES und bei Verbindungen mit VSE/ESA, wenn die Länge des Namens auf 4 Byte beschränkt ist, verwenden Sie MQTP.

Weitere Informationen hierzu finden Sie im Abschnitt [Einstellungen auf dem lokalen IBM i-System für eine ferne Warteschlangenmanagerplattform](#).

### 12 Controllerbeschreibung

Dieser Parameter ist ein Aliasname für den Steuerpunktnamen (oder Knotennamen) des Partnersystems. In diesem Beispiel wird zur Vereinfachung der tatsächliche Name des Partners verwendet.

### 13 Einheit

Dieser Parameter ist ein Aliasname für die logische Einheit (LU) des Partnersystems. In diesem Beispiel wird zur Vereinfachung der LU-Name des Partners verwendet.

### 14 Nebeninformationen

Dieser Parameter ist der Name des CPI-C-Nebeninformationsprofils. Sie geben einen eigenen Namen, bestehend aus 8 Zeichen, an.

## IBM i **Netzattribute finden**

Der lokale Knoten wurde teilweise als Bestandteil der IBM i-Installation konfiguriert. Geben Sie zur Anzeige der aktuellen Netzattribute den Befehl DSPNETA ein.

Sie ändern diese Werte bei Bedarf mit dem Befehl CHGNETA. Zum Anwenden der Änderungen ist unter Umständen ein IPL erforderlich.

```
Display Network Attributes
System: AS400PU
Current system name . . . . . : AS400PU
Pending system name . . . . . :
Local network ID . . . . . : NETID
Local control point name . . . . . : AS400PU
Default local location . . . . . : AS400LU
Default mode . . . . . : BLANK
APPN node type . . . . . : *ENDNODE
Data compression . . . . . : *NONE
Intermediate data compression . . . . . : *NONE
Maximum number of intermediate sessions . . . . . : 200
Route addition resistance . . . . . : 128
Server network ID/control point name . . . . . : NETID NETCP
```

More...  
Press Enter to continue.

F3=Exit F12=Cancel

Stellen Sie sicher, dass die Werte für **ID des lokalen Netzes** (1), **Name des lokalen Steuerpunkts** (2) und **Lokale Standardposition** (3) mit den Werten auf Ihrem Arbeitsblatt übereinstimmen.

## IBM i **Den Wert des Ressourcennamens ermitteln**

Sie ermitteln den Wert des Ressourcennamens, indem Sie WRKHDWRSC TYPE(\*CMN) eingeben und die Eingabetaste drücken.

Die Anzeige "Work with Communication Resources" (Mit DFV-Ressourcen arbeiten) wird angezeigt. Der Wert **Ressourcenname** ist der Token-Ring-Port. In diesem Beispiel lautet er LIN041.

```
Mit Datenübertragungsressourcen arbeiten
System: AS400PU
Geben Sie Optionen ein, und drücken Sie die Eingabetaste
2=Bearbeiten 4=Entfernen 5=Mit Konfigurationsbeschreibung arbeiten
7=Konfigurationsbeschreibung hinzufügen ...
```

```
Konfiguration
Opt Ressource      Beschreibung Typ Beschreibung
CC02                2636 Comm Prozessor
LIN04                2636 LAN Adapter
LIN041      TOKEN-RING 2636 Token-Ring-Port
```

Unten  
F3=Beenden F5=Aktualisieren F6=Drucken F11=Anzeige Ressourcenadresse/Status  
F12=Abbrechen F23=Weitere Optionen

## Eine LU 6.2-Verbindung einrichten

In diesem Abschnitt wird beschrieben, wie Sie eine LU 6.2-Verbindung einrichten.

## Lokalen Knoten konfigurieren

Zum Konfigurieren des lokalen Knotens müssen Sie eine Leitungsbeschreibung erstellen und einen Routing-Eintrag hinzufügen.

### Eine Leitungsbeschreibung erstellen

1. Wenn die Leitungsbeschreibung nicht bereits erstellt wurde, verwenden Sie den Befehl CRTLINTRN.
2. Geben Sie die Werte für **Leitungsbeschreibung** (6) und **Ressourcenname** (7) an.

```
Line Desc (Token-Ring) erstellen (CRTLINTRN)

Auswahl eingeben und Eingabetaste drücken.

Leitungsbeschreibung..... TOKENRINGL      Name
Ressourcenname.....     LIN041              Name, *NWID
NWI-Typ.....             *FR *FR, *ATM
Online bei IPL.....      *YES *YES, *NO
Wartezeit anhängen..... *NOWAIT *NOWAIT, 15 -180 (1 Sekunde)
Maximale Steuereinheiten..... 40 1-256
Angehängter NWI.....     *NONE Name, *NONE

Unten
F3=Beenden   F4=Abfragen   F5=Aktualisieren   F10=Zusätzliche Parameter   F12=Abbrechen
F13=Wie diese Anzeige verwendet wird   F24=Weitere Tasten
Parameter LIND erforderlich. +
```

### Einen Routing-Eintrag hinzufügen

1. Geben Sie den Befehl ADDRTGE ein und drücken Sie die Eingabetaste.

```
Leitwegeintrag hinzufügen (ADDRTGE)

Auswahl eingeben und Eingabetaste drücken.

Subsystembeschreibung.... QCMN-Name
Bibliothek.....         *LIBL-Name, *LIBL, *CURLIB
Die Folgenummer des Leitwegeintrags. 1 1-9999
Vergleichsdaten:
Vergleichswert.....    'MQSERIES'

Startposition.....      37 1-80
Aufrufprogramm.....     AMQCRC6B Name, *RTGDTA
Bibliothek.....         QMAS400      Name, * LI BL, *CURLIB
Klasse.....             *SBSD-Name, *SBSD
Bibliothek.....         *LIBL-Name, *LIBL, *CURLIB
Maximale aktive Routing-Schritte.. *NOMAX 0-1000, *NOMAX
Speicherpool-ID....    1 1-10

Unten
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur
Verwendung dieser Anzeige
F24 = Weitere Tasten
Parameter SBSDB erforderlich. +
```

2. Geben Sie Ihren Wert für **Subsystembeschreibung** (5) sowie die hier angezeigten Werte für **Folgennummer des Routing-Eintrags**, **Vergleichswert** (8), **Anfangsposition**, **Aufzurufendes Programm** und die **Bibliothek** mit dem Programm an, das aufgerufen werden soll.

3. Geben Sie den Befehl STRSBS *Subsystembeschreibung* (5) ein und drücken Sie die Eingabetaste.

## IBM i **Verbindung mit Partnerknoten**

Um eine Verbindung zu einem Partnerknoten herzustellen, müssen Sie: eine Controllerbeschreibung erstellen, eine Einheitenbeschreibung erstellen, CPI-C-Nebeninformationen erstellen, einen DFV-Eintrag für APPC hinzufügen sowie einen Konfigurationslisteneintrag hinzufügen.

Dieses Beispiel zeigt die Verbindung zu einem Windows-System, die Schritte sind jedoch für alle Knoten gleich.

### Eine Controllerbeschreibung erstellen

1. Geben Sie in die Befehlszeile den Befehl CRTCTLAPPC ein und drücken Sie die Eingabetaste.

```

Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

Controller description . . . . . WINNTCP      Name
Link type . . . . . *LAN          *FAX, *FR, *IDLC,
*LAN...
Online at IPL . . . . . *NO        *YES, *NO

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
Parameter CTLD required.          +

```

2. Geben Sie einen Wert für **Controllerbeschreibung** (12) an und setzen Sie **Verbindungstyp** auf \*LAN und **Online bei IPL** auf\*NO.
3. Drücken Sie zweimal die Eingabetaste und anschließend die Taste F10.

```

Create Ctl Desc (APPC) (CRTCTLAPPC)

Type choices, press Enter.

Controller description . . . . . > WINNTCP      Name
Link type . . . . . > *LAN          *FAX, *FR, *IDLC, *LAN...
Online at IPL . . . . . > *NO        *YES, *NO
APPN-capable . . . . . *YES        *YES, *NO
Switched line list . . . . . TOKENRINGL Name
+ for more values
Maximum frame size . . . . . *LINKTYPE  265-16393, 256, 265, 512...
Remote network identifier . . . . NETID      Name, *NETATR, *NONE, *ANY
Remote control point . . . . . WINNTCP      Name, *ANY
Exchange identifier . . . . . 00000000-FFFFFFF
Initial connection . . . . . *DIAL        *DIAL, *ANS
Dial initiation . . . . . *LINKTYPE      *LINKTYPE, *IMMED, *DELAY
LAN remote adapter address . . . 10005AFC5D83 000000000001-FFFFFFFFFFFF
APPN CP session support . . . . *YES        *YES, *NO
APPN node type . . . . . *ENDNODE      *ENDNODE, *LENNODE...
APPN transmission group number  1          1-20, *CALC
More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

4. Geben Sie die Werte für **Wählleitungsliste** (6), **ID des fernen Netzes** (9), **Fernsteuerpunkt** (10) und **Adresse des fernen LAN-Adapters** (16) an.
5. Drücken Sie die Eingabetaste.

## Eine Einheitenbeschreibung erstellen

1. Geben Sie den Befehl CRTDEVAPPCC ein und drücken Sie die Eingabetaste.

```
Create Device Desc (APPC) (CRTDEVAPPCC)

Type choices, press Enter.

Device description . . . . . WINNTLU      Name
Remote location . . . . . WINNTLU      Name
Online at IPL . . . . . *YES          *YES, *NO
Local location . . . . . AS400LU      Name, *NETATR
Remote network identifier . . . . . NETID  Name, *NETATR, *NONE
Attached controller . . . . . WINNTCP   Name
Mode . . . . . *NETATR              Name, *NETATR
+ for more values
Message queue . . . . . QSYSOPR      Name, QSYSOPR
Library . . . . . *LIBL             Name, *LIBL, *CURLIB
APPN-capable . . . . . *YES         *YES, *NO
Single session:
Single session capable . . . . . *NO   *NO, *YES
Number of conversations . . . . . 1-512

Bottom
F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
F13=How to use this display F24=More keys
Parameter DEVD required.          +
```

2. Geben Sie die Werte für **Einheitenbeschreibung** (13), **Ferner Standort** (11), **Lokaler Standort** (3), **ID des fernen Netzes** (9) und **Angehängter Controller** (12) an.

**Anmerkung:** Controller- und Einheitenbeschreibungen müssen nicht manuell erstellt werden, wenn Sie die automatische Konfiguration in IBM i verwenden. Weitere Informationen finden Sie in der Dokumentation zu IBM i.

## CPI-C-Nebeninformationen erstellen

1. Geben Sie den Befehl CRTCSI ein und drücken Sie die Taste F10.

```
Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . NTCPIC      Name
Library . . . . . *CURLIB          Name, *CURLIB
Remote location . . . . . WINNTLU      Name
Transaction program . . . . . MQSERIES

Text 'description' . . . . . *BLANK

Additional Parameters

Device . . . . . *LOC              Name, *LOC
Local location . . . . . AS400LU      Name, *LOC, *NETATR
Mode . . . . . #INTER              Name, *NETATR
Remote network identifier . . . . . NETID  Name, *LOC, *NETATR, *NONE
Authority . . . . . *LIBCRTAUT      Name, *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
Parameter CSI required.
```

2. Geben Sie die Werte für **Nebeninformationen** (14), **Ferner Standort** (11), **Transaktionsprogramm** (15), **Lokaler Standort** (3), **Modus** und **ID des fernen Netzes** (9) an.
3. Drücken Sie die Eingabetaste.

## Einen DFV-Eintrag für APPC hinzufügen

1. Geben Sie in die Befehlszeile den Befehl ADDCMNE ein und drücken Sie die Eingabetaste.

```
Add Communications Entry (ADDCMNE)

Type choices, press Enter.

Subsystem description . . . . . QCMN      Name
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Device . . . . . WINNTLU     Name, generic*, *ALL...
Remote location . . . . .      Name
Job description . . . . . *USRPRF   Name, *USRPRF, *SBSD
Library . . . . .      Name, *LIBL, *CURLIB
Default user profile . . . . . *NONE   Name, *NONE, *SYS
Mode . . . . . *ANY         Name, *ANY
Maximum active jobs . . . . . *NOMAX   0-1000, *NOMAX

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
Parameter SBSD required.
```

2. Geben Sie die Werte für **Subsystembeschreibung** (5) und **Gerät** (13) an und drücken Sie die Eingabetaste.

## Einen Konfigurationslisteneintrag hinzufügen

1. Geben Sie den Befehl ADDCFGLE \*APPNRMT ein und drücken Sie die Taste F4.

```
Add Configuration List Entries (ADDCFGLE)

Type choices, press Enter.

Configuration list type . . . . > *APPNRMT *APPNLCL, *APPNRMT...
APPN remote location entry:
Remote location name . . . . . WINNTLU     Name, generic*, *ANY
Remote network identifier . . . . NETID     Name, *NETATR, *NONE
Local location name . . . . . AS400LU     Name, *NETATR
Remote control point . . . . . WINNTCP    Name, *NONE
Control point net ID . . . . . NETID      Name, *NETATR, *NONE
Location password . . . . . *NONE
Secure location . . . . . *NO           *YES, *NO
Single session . . . . . *NO           *YES, *NO
Locally controlled session . . . *NO           *YES, *NO
Pre-established session . . . . *NO           *YES, *NO
Entry 'description' . . . . . *BLANK
Number of conversations . . . . 10       1-512
+ for more values

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

2. Geben Sie die Werte für **Name des fernen Standorts** (11), **ID des fernen Netzes** (9), **Name des lokalen Standorts** (3), **Fernsteuerpunkt** (10) und **Netz-ID des Kontrollpunkts** (9) an.
3. Drücken Sie die Eingabetaste.

### ► IBM i **Wie geht es weiter?**

Die LU 6.2-Verbindung ist nun eingerichtet. Sie können die Konfiguration abschließen.

Fahren Sie dazu mit „IBM MQ for IBM i-Konfiguration“ auf Seite 25 fort.

Wenn TCP bereits konfiguriert ist, fallen keine zusätzlichen Konfigurationsaufgaben an. Ist TCP/IP nicht konfiguriert, müssen Sie eine TCP/IP-Schnittstelle, eine TCP/IP-Loopback-Schnittstelle und eine Standardroute hinzufügen.

### Eine TCP/IP-Schnittstelle hinzufügen

1. Geben Sie in der Befehlszeile den Befehl ADDTCPIFC ein und drücken Sie die Eingabetaste.

```
TCP/IP-Schnittstelle hinzufügen (ADDTCPICF)
```

```
Auswahl eingeben und Eingabetaste drücken.
```

```
Internetadresse..... 19.22.11.55
Leitungsbeschreibung..... TOKENRINGL      Name, *LOOPBACK
Teilnetzmaske..... 255.255.0.0
Art des Service..... *NORMAL *MINDELAY, *MAXTHRPUT ..
Maximale Übertragungseinheit... *LIND 576-16388, *LIND
Autostart..... *YES *YES, *NO
PVC Kennung des logischen Kanals                001-FFF
+ für mehr Werte
Zeitlimit für inaktive X.25 -Schaltung... 60 1-600
Maximale virtuelle X.25 -Schaltkreise. 64 0-64
X.25 DDN-Schnittstelle..... *NO *YES, *NO
TRLAN-Bitfolgeplanung..... *MSB *MSB, *LSB
```

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige  
F24 = Weitere Tasten

2. Geben Sie die **IP-Adresse**, die **Leitungsbeschreibung** sowie eine **Teilnetzmaske** des Systems an.
3. Drücken Sie die Eingabetaste.

### Eine TCP/IP-Loopback-Schnittstelle hinzufügen

1. Geben Sie in der Befehlszeile den Befehl ADDTCPIFC ein und drücken Sie die Eingabetaste.

```
TCP-Schnittstelle hinzufügen (ADDTCPICF)
```

```
Auswahl eingeben und Eingabetaste drücken.
```

```
Internetadresse..... 127.0.0.1
Leitungsbeschreibung..... *LOOPBACK-Name, *LOOPBACK
Teilnetzmaske..... 255.0.0.0
Art des Service..... *NORMAL *MINDELAY, *MAXTHRPUT ..
Maximale Übertragungseinheit... *LIND 576-16388, *LIND
Autostart..... *YES *YES, *NO
PVC Kennung des logischen Kanals                001-FFF
+ für mehr Werte
Zeitlimit für inaktive X.25 -Schaltung... 60 1-600
Maximale virtuelle X.25 -Schaltkreise. 64 0-64
X.25 DDN-Schnittstelle..... *NO *YES, *NO
TRLAN-Bitfolgeplanung..... *MSB *MSB, *LSB
```

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige  
F24 = Weitere Tasten

2. Geben Sie die Werte für die **IP-Adresse**, die **Leitungsbeschreibung** und die **Teilnetzmaske** an.



## Eine Standardroute hinzufügen

1. Geben Sie in der Befehlszeile den Befehl ADDTCP RTE ein und drücken Sie die Eingabetaste.

```
TCP Routing hinzufügen (ADDTCP RTE)
```

```
Auswahl eingeben und Eingabetaste drücken.
```

```
Route-Ziel..... *DFTRROUTE  
Teilnetzmaske..... *NONE  
Art des Service..... *NORMAL *MINDELAY, *MAXTHRPUT.  
Nächster Hop..... 19.2.3.4  
Maximale Übertragungseinheit... 576 576-16388, *IFC
```

```
Unten
```

```
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur  
Verwendung dieser Anzeige  
F24 = Weitere Tasten  
Eingabeaufforderung wurde beendet, nachdem Benutzer F12 gedrückt hat.
```

2. Geben Sie die Werte für Ihr Netz ein und drücken Sie die Eingabetaste, um einen Standardrouteneintrag zu erstellen.

## Wie geht es weiter?

Die TCP-Verbindung ist nun eingerichtet. Sie können die Konfiguration abschließen. Fahren Sie dazu mit „IBM MQ for IBM i-Konfiguration“ auf Seite 25 fort.

### IBM MQ for IBM i-Konfiguration

Verwenden Sie zum Konfigurieren von IBM MQ for IBM i den Befehl WRKMQMQ, um das Konfigurationsmenü anzuzeigen.

Starten Sie das TCP-Kanal-Empfangsprogramm mit dem Befehl STRMQMLSR.

Starten Sie einen beliebigen Senderkanal mit dem Befehl STRMQMCHL CHLNAME(*Kanalname*).

Mit dem Befehl WRKMQMQ zeigen Sie das Konfigurationsmenü von IBM MQ an.

**Anmerkung:** AMQ\*-Fehler werden im Protokoll in Relation zu dem Job gesetzt, der den Fehler gefunden hat. Mit dem Befehl WRKACTJOB zeigen Sie die Liste der Jobs an. Suchen Sie unter dem Subsystemnamen QSYSWRK nach dem Job und geben Sie 5 ein, um mit diesem Job zu arbeiten. IBM MQ-Protokolle haben das Präfix AMQ.

### **Einen Warteschlangenmanager erstellen**

Führen Sie die folgenden Schritte aus, um einen Warteschlangenmanager der Basiskonfiguration einzurichten.

1. Zunächst müssen Sie einen Warteschlangenmanager erstellen. Geben Sie dazu den Befehl CRTMQM ein und drücken Sie die Eingabetaste.

### Nachrichten-Warteschlangenmanager erstellen (CRTMQM)

Auswahl eingeben und Eingabetaste drücken.

Name des Nachrichten-WS-Managers...

Text 'Beschreibung'..... \*BLANK

Auslöserintervall..... 999999999 0-999999999

Nicht abgegebene Nachrichtenwarteschlange... \*NONE

Standardübertragungswarteschlange... \*NONE

Maximale Griffgrenze..... 256 1-999999999

Maximale nicht festgeschriebene Nachrichten.. 1000 1-10000

Standard-WS-Manager..... \*NO \*YES, \*NO

Unten

F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen  
zur Verwendung dieser Anzeige  
F24 = Weitere Tasten

2. Geben Sie im Feld **Message Queue Manager name** (Name des Nachrichten-Warteschlangenmanagers) den Namen AS400 ein. Geben Sie im Feld **Undelivered message queue** (Warteschlange für nicht zugestellte Nachrichten) DEAD.LETTER.QUEUE ein.
3. Drücken Sie die Eingabetaste.
4. Starten Sie nun den Warteschlangenmanager, indem Sie den Befehl STRMQM MQMNAME (AS400) eingeben.
5. Erstellen Sie die Warteschlange für nicht zugestellte Nachrichten unter Verwendung der folgenden Parameter. (Ausführliche Informationen und ein Beispiel hierzu finden Sie unter „[Eine Warteschlange definieren](#)“ auf Seite 26.)

```
Local Queue
Queue name :  DEAD.LETTER.QUEUE
Queue type :  *LCL
```

### **Eine Warteschlange definieren**

Mit dem Befehl CRTMQMQ können Sie eine Warteschlange definieren.

Geben Sie in der Befehlszeile den Befehl CRTMQMQ ein.

```
MQM-Warteschlange erstellen (CRTMQMQ)
Auswahl eingeben und Eingabetaste drücken.
Warteschlangenname.....
Warteschlangentyp.....      *ALS, *LCL, *RMT
```

Unten  
F3 = Verlassen F4 = Bedienerführung F5 = Aktualisieren F12 = Abbrechen F13 = Informationen zur Verwendung dieser Anzeige  
F24 = Weitere Tasten  
Parameter QNAME erforderlich.

Füllen Sie die beiden Felder in dieser Anzeige aus und drücken Sie die Eingabetaste. Daraufhin wird eine weitere Anzeige mit Eingabefeldern für Ihre weiteren Parameter eingeblendet. Für alle anderen Warteschlangenattribute können Sie die Standardeinstellungen übernehmen.

### **Kanal unter IBM i definieren**

Unter IBM i können Sie einen Kanal mit dem Befehl CRTMQMCHL definieren.

Geben Sie in der Befehlszeile den Befehl CRTMQMCHL ein.

```
Create MQM Channel (CRTMQMCHL)
Type choices, press Enter.
Channel name . . . . .
Channel type . . . . .      *RCVR, *SDR, *SVR, *RQSTR
```

Bottom  
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display  
F24=More keys  
Parameter CHLNAME required.

Füllen Sie die beiden Felder in dieser Anzeige aus und drücken Sie die Eingabetaste. Daraufhin wird eine weitere Anzeige eingeblendet, in der Sie die Werte für die zuvor angegebenen Parameter festlegen können. Für alle anderen Kanalattribute können Sie die Standardeinstellungen übernehmen.

### **Kanalkonfiguration für IBM i**

Sie müssen Ihre Kanäle konfigurieren, um die Beispielkonfigurationskanäle zu implementieren.

In diesem Abschnitt wird die Konfiguration des IBM i-Warteschlangenmanagers zur Implementierung des in „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for IBM i und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Werte im Beispiel für Windows ein.

**Anmerkung:**

1. Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.
2. Der Pingbefehl für den IBM MQ-Kanal (PNGMQMCHL) wird interaktiv ausgeführt, während das Starten eines Kanals die Übergabe eines Batch-Jobs auslöst. Wenn der Pingbefehl eines Kanals erfolgreich ausgeführt wird, der Kanal jedoch nicht startet, sind möglicherweise das Netz und die IBM MQ-Definitionen korrekt, nicht jedoch die IBM i-Umgebung für den Batch-Job. Stellen Sie beispielsweise sicher, dass QSYS2 nicht nur in Ihre persönliche Bibliotheksliste, sondern auch in den Systemteil der Bibliotheksliste aufgenommen wird.

Ausführliche Informationen und Beispiele zum Erstellen der aufgeführten Objekte finden Sie unter „Eine Warteschlange definieren“ auf Seite 26 und „Kanal unter IBM i definieren“ auf Seite 27.






<i>Tabelle 3. Konfigurationsbeispiele für IBM i</i>			
<b>ID</b>	<b>Parameter Name (Parametername)</b>	<b>Referenz</b>	<b>Verwendetes Beispiel</b>
<b>Definition für lokalen Knoten</b>			
A	Name des Warteschlangenmanagers		<b>AS400</b>
B	Name der lokalen Warteschlange		<b>AS400.LOCALQ</b>
  <b>Verbindung zu IBM MQ for Windows</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Windows“ auf Seite 46 verwendeten Werten wie angegeben entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>WINNT</b>
G	Name des Senderkanals (SNA)		<b>AS400.WINNT.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>AS400.WINNT.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.AS400.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>WINNT.AS400.TCP</b>
  <b>Verbindung zu IBM MQ for AIX</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für AIX“ auf Seite 11 verwendeten Werten wie angegeben entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>
D	Name der fernen Warteschlange		<b>AIXREMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>

Tabelle 3. Konfigurationsbeispiele für IBM i (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
F	Name der Übertragungswarteschlange		<b>AIX</b>
G	Name des Senderkanals (SNA)		<b>AS400.AIX.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>AS400.AIXTCP-</b>
I	Name des Empfängerkanals (SNA)	G	<b>AIX.AS400.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>AIX.AS400.TCP</b>
<p><b>Solaris</b> <b>Solaris</b> <b>Verbindung zu IBM MQ for Solaris</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 39 verwendeten Werten wie angegeben entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>Solaris</b>
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>Solaris</b>
G	Name des Senderkanals (SNA)		<b>AS400.SOLARIS.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>AS400.SOLARIS.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.AS400.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.AS400.TCP</b>
<p><b>Linux</b> <b>Linux</b> <b>Verbindung zu IBM MQ for Linux</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 34 verwendeten Werten wie angegeben entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>LINUX</b>
G	Name des Senderkanals (SNA)		<b>AS400.LINUX.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>AS400.LINUX.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.AS400.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.AS400.TCP</b>
<p><b>z/OS</b> <b>z/OS</b> <b>Verbindung zu IBM MQ for z/OS</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für z/OS“ auf Seite 51 verwendeten Werten wie angegeben entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>MVS</b>

Tabelle 3. Konfigurationsbeispiele für IBM i (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
G	Name des Senderkanals (SNA)		<b>AS400.MVS.SNA</b>
H	Name des Senderkanals (TCP)		<b>AS400.MVS.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>MVS.AS400.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>MVS.AS400.TCP</b>
<b>Verbindung zu MQSeries for VSE/ESA</b>			
Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>VSE</b>
D	Name der fernen Warteschlange		<b>VSE.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>VSE.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>VSE</b>
G	Name des Senderkanals		<b>AS400.VSE.SNA</b>
I	Name des Empfängerkanals	G	<b>VSE.AS400.SNA</b>

 **Senderkanaldefinitionen für IBM i**  
Beispiel für Senderkanaldefinitionen für SNA und TCP.

### SNA verwenden

```

Local Queue
  Queue name :   WINNT
  Queue type  :   *LCL
  Usage      :   *TMQ
                                     F

Remote Queue
  Queue name :   WINNT.REMOTEQ
  Queue type  :   *RMT
  Remote queue :   WINNT.LOCALQ
  Remote Queue Manager :   WINNT
  Transmission queue :   WINNT
                                     C
                                     F

Sender Channel
  Channel Name :   AS400.WINNT.SNA
  Channel Type  :   *SDR
  Transport type :   *LU62
  Connection name :   WINNTCPIC
  Transmission queue :   WINNT
                                     14
                                     F
    
```

### TCP verwenden

```

Local Queue
  Queue name :   WINNT
  Queue type  :   *LCL
  Usage      :   *TMQ
                                     F

Remote Queue
  Queue name :   WINNT.REMOTEQ
  Queue type  :   *RMT
  Remote queue :   WINNT.LOCALQ
  Remote Queue Manager :   WINNT
  Transmission queue :   WINNT
                                     C
                                     F

Sender Channel
  Channel Name :   AS400.WINNT.TCP
  Channel Type  :   *SDR
                                     H
    
```

```
Transport type : *TCP
Connection name : WINNT.tcpip.hostname
Transmission queue : WINNT
```

## IBM i Empfängerkanaldefinitionen für IBM i

Beispiel für Empfängerkanaldefinitionen für SNA und TCP.

### SNA verwenden

```
Local Queue
Queue name : AS400.LOCALQ      B
Queue type : *LCL

Receiver Channel
Channel Name : WINNT.AS400.SNA  I
Channel Type : *RCVR
Transport type : *LU62
```

### TCP verwenden

```
Local Queue
Queue name : AS400.LOCALQ      B
Queue type : *LCL

Receiver Channel
Channel Name : WINNT.AS400.TCP  J
Channel Type : *RCVR
Transport type : *TCP
```

## Linux IBM MQ-Beispielkonfiguration für Linux

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for Linux- und IBM MQ-Produkten.

Für folgende Plattformen stehen Beispiele zur Verfügung:

- **Windows** Windows
- **AIX** AIX
- **Solaris** Solaris
- **IBM i** IBM i
- **z/OS** z/OS
- VSE/ESA

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## Linux Eine LU 6.2-Verbindung einrichten

Verwenden Sie dieses Arbeitsblatt, um die Werte für Ihre Konfiguration zu erfassen.

**Anmerkung:** Die Informationen in diesem Abschnitt gelten nur für IBM MQ for Linux (x86-Plattform). Sie gelten nicht für IBM MQ for Linux (x86-64-Plattform), IBM MQ for Linux (zSeries s390x-Plattform) und IBM MQ for Linux (Power-Plattform).

Aktuelle Informationen zur Konfiguration von SNA über TCP/IP finden Sie im Administrationshandbuch für Ihre Version von Linux in der folgenden Dokumentation: [Communications Server for Linux library](#).

## Linux TCP-Verbindung unter Linux aufbauen

In einigen Linux-Distributionen wird jetzt der erweiterte inet-Dämon (XINETD) anstelle des inet-Dämons (INETD) verwendet. Die folgenden Anweisungen dienen zum Aufbauen einer TCP-Verbindung mit dem inet-Dämon oder dem erweiterten inet-Dämon.

### inet-Dämon (INETD) verwenden

`MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

Gehen Sie wie folgt vor, um eine TCP-Verbindung aufzubauen.

1. Bearbeiten Sie die Datei `"/etc/services"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries    1414/tcp    # MQSeries channel listener
```

**Anmerkung:** Zur Bearbeitung dieser Datei müssen Sie als Superuser oder Rootbenutzer angemeldet sein.

2. Bearbeiten Sie die Datei `"/etc/inetd.conf"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m queue.manager.name ]
```

3. Ermitteln Sie die Prozess-ID von InetD durch Eingabe folgenden Befehls:

```
ps -ef | grep inetd
```

4. Führen Sie folgenden Befehl aus:

```
kill -1 inetd processid
```

Wenn Sie mehrere Warteschlangenmanager in Ihrem System betreiben und demzufolge mehr als einen Service benötigen, müssen Sie für jeden einzelnen Warteschlangenmanager sowohl in `"/etc/services"` als auch in `"inetd.conf"` eine weitere Zeile hinzufügen.

Beispiel:

```
MQSeries1    1414/tcp  
MQSeries2    1822/tcp
```

```
MQSeries1 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1  
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

So lässt sich vermeiden, dass Fehlermeldungen ausgegeben werden, wenn eine Beschränkung für die Anzahl der ausstehenden Verbindungsanforderungen gilt, die in der Warteschlange eines einzelnen TCP-Ports stehen können. Informationen zur Anzahl der ausstehenden Verbindungsanforderungen finden Sie im Abschnitt [Rückstandsoption des TCP-Empfangsprogramms verwenden](#).

Der InetD-Prozess kann unter Linux die Rate der an einem TCP-Anschluss eingehenden Verbindungen einschränken. Die Standardrate sind 40 Verbindungen mit einem Intervall von 60 Sekunden. Wenn Sie eine höhere Rate benötigen, geben Sie eine neue Beschränkung für die Anzahl der eingehenden Verbindungen in einem 60-Sekunden-Intervall an, indem Sie einen Punkt (.) gefolgt von dem neuen Grenzwert



an den Parameter 'nowait' des betreffenden Service in 'inetd.conf' anhängen. Geben Sie beispielsweise für einen Grenzwert von 500 Verbindungen in einem 60-Sekunden-Intervall Folgendes ein:

```
MQSeries stream tcp nowait.500 mqm / MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM1
```

`MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

## Den erweiterten inet-Dämon (XINETD) verwenden

Die folgenden Anweisungen beschreiben, wie der erweiterte inet-Dämon in Red Hat Linux-Systeme implementiert wird. Wenn Sie eine andere Linux-Distribution verwenden, müssen diese Anweisungen gegebenenfalls entsprechend angepasst werden.

Gehen Sie wie folgt vor, um eine TCP-Verbindung aufzubauen.

1. Bearbeiten Sie die Datei "/etc/services". Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries 1414/tcp # MQSeries channel listener
```

**Anmerkung:** Zur Bearbeitung dieser Datei müssen Sie als Superuser oder Rootbenutzer angemeldet sein.

2. Erstellen Sie im XINETD-Konfigurationsverzeichnis "/etc/xinetd.d" eine Datei mit dem Namen IBM MQ. Fügen Sie die folgende Zeilengruppe zur Datei hinzu:

```
# IBM MQ service for XINETD
service MQSeries
{
  disable          = no
  flags            = REUSE
  socket_type     = stream
  wait            = no
  user            = mqm
  server          = MQ_INSTALLATION_PATH/bin/amqcrsta
  server_args     = -m queue.manager.name
  log_on_failure += USERID
}
```

3. Starten Sie den erweiterten inet-Dämon erneut, indem Sie den folgenden Befehl eingeben:

```
/etc/rc.d/init.d/xinetd restart
```

Wenn Sie mehrere Warteschlangenmanager in Ihrem System betreiben und demzufolge mehr als einen Service benötigen, müssen Sie für jeden einzelnen Warteschlangenmanager in der Datei '/etc/services' eine weitere Zeile hinzufügen. Sie können für die einzelnen Services im Verzeichnis '/etc/xinetd.d' eine Datei erstellen. Alternativ können Sie der zuvor erstellten IBM MQ-Datei weitere Zeilengruppen hinzufügen.

Der xinetd-Prozess kann unter Linux die Rate der an einem TCP-Anschluss eingehenden Verbindungen einschränken. Der Standardwert beträgt 50 Verbindungen in einem 10-Sekunden-Intervall. Wenn Sie eine höhere Rate benötigen, geben Sie für den Rate der eingehenden Verbindungen einen Grenzwert an, indem Sie in der xinetd-Konfigurationsdatei das Attribut "cps" angeben. Geben Sie beispielsweise für einen Grenzwert von 500 Verbindungen in einem 60-Sekunden-Intervall Folgendes ein:

```
cps = 500 60
```

## Wie geht es weiter?

Die TCP/IP-Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Wechseln Sie zu „Konfiguration von IBM MQ for Linux“ auf Seite 34.

Bevor Sie mit dem Installationsprozess beginnen, stellen Sie sicher, dass die Benutzer-ID und die Gruppe "mqm" erstellt und das Kennwort festgelegt wurden.

Starten Sie einen beliebigen Kanal über den Befehl:

```
runmqchl -c channel.name
```

**Anmerkung:**

1. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert, wobei `MQ_INSTALLATION_PATH` das übergeordnete Verzeichnis angibt, in dem IBM MQ installiert ist.
2. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.
3. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

**Basiskonfiguration**

1. Erstellen Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
crtmqm -u dlqname -q linux
```

Dabei gilt:

**linux**

ist der Name des Warteschlangenmanagers

**-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

**-u dlqname**

gibt den Namen der Warteschlange für nicht zustellbare Nachrichten an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

2. Starten Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
strmqm linux
```

, hierbei ist `linux` der Name, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

Im folgenden Abschnitt finden Sie ausführliche Informationen zu der Konfiguration des Linux-Warteschlangenmanagers für die Implementierung des in [„IBM MQ-Beispielkonfiguration für alle Plattformen“](#) auf Seite 5 beschriebenen Kanals.

Es ist der MQSC-Befehl zum Erstellen der einzelnen Objekte angegeben. Entweder starten Sie **runmqsc** in einer UNIX-Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die angegebenen Beispiele gelten für die Verbindung zwischen IBM MQ for Linux.

**Anmerkung:** Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.

Tabelle 4. Konfigurationsbeispiele für IBM MQ for Linux







ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<b>Definition für lokalen Knoten</b>			
A	Name des Warteschlangenmanagers		<b>LINUX</b>
B	Name der lokalen Warteschlange		<b>LINUX.LOCALQ</b>
  <b>Verbindung zu IBM MQ for Windows</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für Windows</a> “ auf Seite 46 angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>WINNT</b>
G	Name des Senderkanals (SNA)		<b>LINUX.WINNT.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>LINUX.WINNT.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.LINUX.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.LINUX.TCP</b>
  <b>Verbindung zu IBM MQ for AIX</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für AIX</a> “ auf Seite 11 angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>
D	Name der fernen Warteschlange		<b>AIXREMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>AIX</b>
G	Name des Senderkanals (SNA)		<b>LINUX.AIX.SNA</b>
H	Name des Senderkanals (TCP)		<b>LINUX.AIX.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>AIX.LINUX.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>AIX.LINUX.TCP</b>
  <b>Verbindung zu IBM MQ for Solaris</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 5</a> auf Seite 39 angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>Solaris</b>
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>GIS</b>
G	Name des Senderkanals (SNA)		<b>LINUX.SOLARIS.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>LINUX.SOLARIS.TCP</b>

Tabelle 4. Konfigurationsbeispiele für IBM MQ for Linux (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.LINUX.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.LINUX.TCP</b>
<div style="background-color: #004a99; color: white; padding: 2px; display: inline-block;">IBM i</div> <div style="background-color: #004a99; color: white; padding: 2px; display: inline-block;">IBM i</div> <b>Verbindung zu IBM MQ for IBM i</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 3 auf Seite 28</a> angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>AS400</b>
G	Name des Senderkanals (SNA)		<b>LINUX.AS400.SNA</b>
H	Name des Senderkanals (TCP)		<b>LINUX.AS400.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>AS400.LINUX.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>AS400.LINUX.TCP</b>
<div style="background-color: #c00000; color: white; padding: 2px; display: inline-block;">z/OS</div> <div style="background-color: #c00000; color: white; padding: 2px; display: inline-block;">z/OS</div> <b>Verbindung zu IBM MQ for z/OS</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 7 auf Seite 51</a> angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>MVS</b>
G	Name des Senderkanals (SNA)		<b>LINUX.MVS.SNA</b>
H	Name des Senderkanals (TCP)		<b>LINUX.MVS.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>MVS.LINUX.SNA</b>

**Linux** *Senderkanaldefinitionen mithilfe der SNA in IBM MQ for Linux (x86-Plattform)*

Mustercode:

```

def q1 (HPUX) +                               F
  usage(xmitq) +
  replace

def qr (HPUX.REMOTEQ) +                       D
  rname(HPUX.LOCALQ) +                       E
  rqnname(HPUX) +                             C
  xmitq(HPUX) +                               F
  replace

def chl (LINUX.HPUX.SNA) chltype(sdr) +       G
  trptype(lu62) +
  conname('HPUXCPIC') +                     14
  xmitq(HPUX) +                               F
  replace
  
```

**Linux** *Empfängerkanaldefinitionen mithilfe der SNA in IBM MQ for Linux (x86-Plattform)*

Mustercode:

```
def ql (LINUX.LOCALQ) replace B
def chl (HPUX.LINUX.SNA) chltype(rcvr) + I
  trptype(lu62) +
  replace
```

### Linux *Senderkanaldefinitionen über TCP in IBM MQ for Linux*

Mustercode:

```
def ql (HPUX) + F
  usage(xmitq) +
  replace

def qr (HPUX.REMOTEQ) + D
  rname(HPUX.LOCALQ) + E
  rqmname(HPUX) + C
  xmitq(HPUX) + F
  replace

def chl (LINUX.HPUX.TCP) chltype(sdr) + H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(HPUX) + F
  replace
```

### Linux *Empfängerkanaldefinitionen über TCP/IP in IBM MQ for Linux*






Mustercode:

```
def ql (LINUX.LOCALQ) replace B
def chl (HPUX.LINUX.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

## Solaris **IBM MQ-Beispielkonfiguration für Solaris**

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for Solaris- und IBM MQ-Produkten.

Für folgende Plattformen stehen Beispiele zur Verfügung:

-  Windows
-  AIX
-  Linux
-  IBM i
-  z/OS
- VSE/ESA

Unter „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

### Solaris **LU 6.2-Verbindung mit SNAP-IX aufbauen**

Parameter für die Konfiguration einer LU 6.2-Verbindung mit SNAP-IX.

Aktuelle Informationen zur Konfiguration von SNA über TCP/IP finden Sie in der IBM Onlinedokumentation unter [Communications Server](#) sowie in der MetaSwitch-Onlinedokumentation [SNAP-IX Administration Guide](#) und in der Sun-Onlinedokumentation [Configuring Intersystem Communications \(ISC\)](#).

## Solaris Eine TCP-Verbindung einrichten

Informationen zur Konfiguration einer TCP-Verbindung und weitere Schritte.

Gehen Sie wie folgt vor, um eine TCP-Verbindung aufzubauen.

1. Bearbeiten Sie die Datei `"/etc/services"`.

**Anmerkung:** Hierzu müssen Sie sich als Superuser oder Root-Benutzer anmelden. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries      1414/tcp      # MQSeries channel listener
```

2. Bearbeiten Sie die Datei `"/etc/inetd.conf"`. Wenn die folgende Zeile nicht in dieser Datei enthalten ist, fügen Sie sie wie dargestellt hinzu:

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrista amqcrista  
[-m queue.manager.name]
```

`MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

3. Ermitteln Sie die Prozess-ID von InetD durch Eingabe folgenden Befehls:

```
ps -ef | grep inetd
```

4. Führen Sie den entsprechenden Befehl aus:

- Für Solaris 9:

```
kill -1 inetd processid
```

- Für Solaris 10 oder höher:

```
inetconv
```

## Wie geht es weiter?

Die TCP/IP-Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen. Wechseln Sie zu [„IBM MQ for Solaris-Konfiguration“](#) auf Seite 38.

## Solaris IBM MQ for Solaris-Konfiguration

Beschreibt die zum Abschließen der Konfiguration zu definierenden Kanäle.

Bevor Sie mit dem Installationsprozess beginnen, stellen Sie sicher, dass der Benutzer und die Gruppe `mqm` erstellt und das Kennwort festgelegt wurden.

Starten Sie einen beliebigen Kanal über den Befehl:

```
runmqchl -c channel.name
```

### Anmerkung:

1. Beispielprogramme sind im Verzeichnis `MQ_INSTALLATION_PATH/samp` installiert.  
`MQ_INSTALLATION_PATH` steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.
2. Fehlerprotokolle werden unter `"/var/mqm/qmgrs/qmgrname/errors"` gespeichert.
3. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

4. Wenn bei einem SNA- oder LU 6.2-Kanal bei dem Versuch, die Kommunikationsbibliothek zu laden, ein Fehler auftritt, ist eventuell die Datei "liblu62.so" nicht auffindbar. Dieses Problem lässt sich möglicherweise durch Hinzufügen der Speicherposition dieser Datei, meist ist dies das Verzeichnis "/opt/SUNWlu62", zu LD\_LIBRARY\_PATH lösen.

## Basiskonfiguration

1. Erstellen Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
crtmqm -u dlqname -q solaris
```

Dabei gilt:

**solaris**

ist der Name des Warteschlangenmanagers

**-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

**-u dlqname**

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

2. Starten Sie den Warteschlangenmanager in der UNIX-Eingabeaufforderung mithilfe des folgenden Befehls:

```
strmqm solaris
```

, wobei *solaris* der Name ist, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

### **Kanalkonfiguration für Solaris**

Im folgenden Abschnitt finden Sie ausführliche Informationen zur Konfiguration des Solaris-Warteschlangenmanagers für die Implementierung eines Kanals.

Die beschriebene Konfiguration dient zur Implementierung des in [Abbildung 1 auf Seite 5](#) beschriebenen Kanals.

Es ist der MQSC-Befehl zum Erstellen der einzelnen Objekte angegeben. Entweder starten Sie **runmqsc** in einer UNIX-Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for Solaris und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werten im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettdruck** sind benutzerspezifisch und geben die Namen von IBM MQ-Objekten an, die in diesen Beispielen verwendet werden. Wenn Sie die hier verwendeten Namen ändern, müssen Sie auch die anderen Verweise in diesem Abschnitt auf diese Objekte ändern. Alle anderen sind Schlüsselwörter und müssen wie dargestellt eingegeben werden.

Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris				
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
<b>Definition für lokalen Knoten</b>				
A	Name des Warteschlangenmanagers		<b>Solaris</b>	

Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
B	Name der lokalen Warteschlange		<b>SOLARIS.LOCALQ</b>	
<div style="background-color: #800000; color: white; padding: 2px; display: inline-block;">Windows</div> <div style="background-color: #800000; color: white; padding: 2px; display: inline-block; margin-left: 10px;">Windows</div> <b>Verbindung zu IBM MQ for Windows</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 6 auf Seite 46</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>	
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>WINNT</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.WINNT.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>SOLARIS.WINNT.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>WINNT.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>WINNT.SOLARIS.TCP</b>	
<div style="background-color: #008000; color: white; padding: 2px; display: inline-block;">AIX</div> <div style="background-color: #008000; color: white; padding: 2px; display: inline-block; margin-left: 10px;">AIX</div> <b>Verbindung zu IBM MQ for AIX</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 1 auf Seite 12</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>	
D	Name der fernen Warteschlange		<b>AIXREMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AIX</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.AIX.SNA</b>	
H	Name des Senderkanals (TCP)		<b>SOLARIS.AIX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AIX.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AIX.SOLARIS.TCP</b>	
<div style="background-color: #008000; color: white; padding: 2px; display: inline-block;">Linux</div> <div style="background-color: #008000; color: white; padding: 2px; display: inline-block; margin-left: 10px;">Linux</div> <b>Verbindung zu IBM MQ for Linux</b>				
Die Werte in diesem Abschnitt der Tabelle müssen den in <a href="#">Tabelle 4 auf Seite 35</a> angegebenen entsprechen.				
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>	
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>	



Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
F	Name der Übertragungswarteschlange		<b>LINUX</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.LINUX.SNA</b>	
H	Name des Senderkanals (TCP/IP)		<b>SOLARIS.LINUX.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.SOLARIS.TCP</b>	

**IBM i** **IBM i** **Verbindung zu IBM MQ for IBM i**

Die Werte in diesem Abschnitt der Tabelle müssen den in [Tabelle 3 auf Seite 28](#) angegebenen entsprechen.

C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>	
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>AS400</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.AS400.SNA</b>	
H	Name des Senderkanals (TCP)		<b>SOLARIS.AS400.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>AS400.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>AS400.SOLARIS.TCP</b>	

**z/OS** **z/OS** **Verbindung zu IBM MQ for z/OS**

Die Werte in diesem Abschnitt der Tabelle müssen den in [Tabelle 7 auf Seite 51](#) angegebenen entsprechen.

C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>	
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>	
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>	
F	Name der Übertragungswarteschlange		<b>MVS</b>	
G	Name des Senderkanals (SNA)		<b>SOLARIS.MVS.SNA</b>	
H	Name des Senderkanals (TCP)		<b>SOLARIS.MVS.TCP</b>	
I	Name des Empfängerkanals (SNA)	G	<b>MVS.SOLARIS.SNA</b>	
J	Name des Empfängerkanals (TCP)	H	<b>MVS.SOLARIS.TCP</b>	

**Verbindung zu MQSeries for VSE/ESA**

Die Werte in diesem Tabellenabschnitt müssen den in Ihrem VSE/ESA-System verwendeten entsprechen.

Tabelle 5. Arbeitsblatt zur Konfiguration für IBM MQ for Solaris (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel	Benutzerwert
C	Name des fernen Warteschlangenmanagers	A	VSE	
D	Name der fernen Warteschlange		VSE.REMOTEQ	
E	Name der Warteschlange auf fernem System	B	VSE.LOCALQ	
F	Name der Übertragungswarteschlange		VSE	
G	Name des Senderkanals		SOLARIS.VSE.SNA	
I	Name des Empfängerkanals	G	VSE.SOLARIS.SNA	

**Solaris** *Senderkanaldefinitionen unter Verwendung der SNAP-IX SNA in IBM MQ for Solaris*

Mustercode:

```
def ql (WINNT) +                                     F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                             D
  rname(WINNT.LOCALQ) +                             E
  rqmname(WINNT) +                                   C
  xmitq(WINNT) +                                     F
  replace

def chl (SOLARIS.WINNT.SNA) chltype(sdr) +          G
  trptype(lu62) +
  conname('NTCPIC') +                               14
  xmitq(WINNT) +                                     F
  replace
```

**Solaris** *Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for Solaris*

Mustercode:

```
def ql (SOLARIS.LOCALQ) replace                     B

def chl (WINNT.SOLARIS.SNA) chltype(rcvr) +        I
  trptype(lu62) +
  replace
```

**Solaris** *Senderkanaldefinitionen über TCP in IBM MQ for Solaris*

Mustercode:

```
def ql (WINNT) +                                     F
  usage(xmitq) +
  replace

def qr (WINNT.REMOTEQ) +                             D
  rname(WINNT.LOCALQ) +                             E
  rqmname(WINNT) +                                   C
  xmitq(WINNT) +                                     F
  replace

def chl (SOLARIS.WINNT.TCP) chltype(sdr) +         H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
```

```
xmitq(WINNT) +  
replace
```

F

## **Solaris** Empfängerkanaldefinitionen über TCP/IP in IBM MQ for Solaris

Mustercode:

```
def q1 (SOLARIS.LOCALQ) replace B  
def chl (WINNT.SOLARIS.TCP) chltype(rcvr) + J  
trptype(tcp) +  
replace
```

## **Windows** IBM MQ-Beispielkonfiguration für Windows

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for Windows- und IBM MQ-Produkten auf anderen Plattformen.

Die Einrichtung von Kommunikationsverbindungen wird für folgende Plattformen erläutert:

- **AIX** AIX
- **Solaris** Solaris
- **Linux** Linux
- **IBM i** IBM i
- **z/OS** z/OS
- VSE/ESA

Sobald die Verbindung eingerichtet ist, müssen Sie einige Kanäle definieren, um die Konfiguration abzuschließen. Im Abschnitt „IBM MQ for Windows-Konfiguration“ auf Seite 45 finden Sie eine Beschreibung der Beispielprogramme und Befehle.

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## **Windows** Eine LU 6.2-Verbindung einrichten

Verweis auf Informationen zum Konfigurieren von AnyNet SNA über TCP/IP.

Aktuelle Informationen zur Konfiguration von AnyNet SNA über TCP/IP finden Sie online in der folgenden IBM Dokumentation: [AnyNet SNA over TCP/IP](#), [SNA Node Operations](#), and [Communications Server for Windows](#)

## **Windows** Eine TCP-Verbindung einrichten

Der TCP-Stapel, der mit Windows-Systemen ausgeliefert wird, umfasst keinen *inet*- bzw. funktional entsprechenden Dämon.

Der IBM MQ-Befehl, über den das IBM MQ for TCP-Empfangsprogramm gestartet wird, lautet:

```
runmqclsr -t tcp
```

Das Empfangsprogramm muss unbedingt vor jeglichen Kanälen gestartet werden. Es ermöglicht empfangenden Kanälen das automatische Starten nach dem Erhalt einer Anforderung von einem Senderkanal für eingehende Nachrichten.

## Wie geht es weiter?

Sobald die TCP/IP-Verbindung besteht, können Sie die Konfiguration abschließen. Fahren Sie dazu mit „IBM MQ for Windows-Konfiguration“ auf Seite 45 fort.

### Windows NetBIOS-Verbindung aufbauen

Eine NetBIOS-Verbindung wird aus einem Warteschlangenmanager initialisiert, der den Parameter "ConnectionName" für seine Kanaldefinition verwendet, um eine Verbindung zu einem Zielempfangsprogramm herzustellen.

Gehen Sie wie folgt vor, um eine NetBIOS-Verbindung einzurichten:

1. Geben Sie an beiden Enden des Kanals in der Konfigurationsdatei des Warteschlangenmanagers "qm.ini" den lokalen NetBIOS-Namen an, den die IBM MQ-Kanalprozesse verwenden sollen. Die NETBIOS-Zeilengruppe in Windows kann an der Sendeseite wie folgt aussehen:

```
NETBIOS:  
LocalName=WNTNETB1
```

und an der Empfangsseite:

```
NETBIOS:  
LocalName=WNTNETB2
```

Die IBM MQ-Prozesse müssen unterschiedliche lokale NetBIOS-Namen verwenden. Verwenden Sie nicht den Systemnamen als NetBIOS-Namen, da dieser bereits von Windows verwendet wird.

2. Überprüfen Sie an beiden Kanalenden die auf Ihrem System verwendete LAN-Adapternummer. Die Standardeinstellung von IBM MQ for Windows für die logische Adapternummer 0 ist, dass NetBIOS über ein IP-Netz ausgeführt wird. Um das systemeigene NetBIOS zu verwenden, müssen Sie die Nummer des logischen Adapters auf 1 setzen. Informationen hierzu finden Sie im Abschnitt [LAN-Adapternummer erstellen](#).

Geben Sie die richtige LAN-Adapternummer in der NETBIOS-Zeilengruppe der Windows-Registrierungsdatenbank an. Beispiel:

```
NETBIOS:  
AdapterNum=1
```

3. Damit der Senderkanal einwandfrei gestartet werden kann, geben Sie den lokalen NetBIOS-Namen in Form der MQNAME-Umgebungsvariablen an:

```
SET MQNAME=WNTNETB1I
```

Dieser Name muss eindeutig sein.

4. Definieren Sie an der Sendeseite einen Kanal mit dem von der anderen Kanalseite verwendeten NetBIOS-Namen. Beispiel:

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(SDR) +  
TRPTYPE(NETBIOS) +  
CONNNAME(WNTNETB2) +  
XMITQ(OS2) +  
MCTYPE(THREAD) +  
REPLACE
```

Die Option MCTYPE(THREAD) muss angegeben werden, weil unter Windows die Senderkanäle als Threads ausgeführt werden müssen.

5. Definieren Sie an der Empfangsseite den zugehörigen Empfängerkanal. Beispiel:

```
DEFINE CHANNEL (WINNT.OS2.NET) CHLTYPE(RCVR) +  
TRPTYPE(NETBIOS) +  
REPLACE
```

6. Starten Sie den Kanalinitiator, da jeder neue Kanal als Thread und nicht als neuer Prozess gestartet wird.

```
runmqchi
```

7. Starten Sie auf der Empfangsseite das Empfangsprogramm von IBM MQ:

```
runmqlsr -t netbios
```

Sie haben die Möglichkeit, Werte für den Warteschlangenmanagernamen, den lokalen NetBIOS-Namen, die Anzahl der Sitzungen, die Anzahl der Namen und die Anzahl der Befehle anzugeben. Weitere Informationen zum Einrichten der NetBIOS-Verbindungen finden Sie im Abschnitt [NetBIOS-Verbindung unter Windows einrichten](#).

## Windows IBM MQ for Windows-Konfiguration

Beispielprogramme und -befehle für die Konfiguration.

### Anmerkung:

1. Mithilfe des Beispielprogramms AMQSBCG können Sie die Inhalte und Header aller Nachrichten in einer Warteschlange anzeigen. Beispiel:

```
AMQSBCG q_name qmgr_name
```

zeigt die Inhalte der Warteschlange *q\_name* an, die im Warteschlangenmanager *qmgr\_name* definiert ist.

Alternativ können Sie den Nachrichten-Browser im IBM MQ Explorer benutzen.

2. Über die Eingabeaufforderung kann jeder Kanal durch folgenden Befehl gestartet werden:

```
runmqchl -c channel.name
```

3. Fehlerprotokolle finden Sie in den Verzeichnissen *MQ\_INSTALLATION\_PATH\qmgrs\qmgrname\errors* und *MQ\_INSTALLATION\_PATH\qmgrs\@system\errors*. In beiden Verzeichnissen befinden sich die neuesten Nachrichten am Ende der Datei "amqerr01.log".

*MQ\_INSTALLATION\_PATH* steht für das übergeordnete Verzeichnis, in dem IBM MQ installiert ist.

4. Wenn Sie Verwaltungsbefehle mithilfe des Befehlsinterpreters **runmqsc** eingeben, zeigt "a +" am Ende einer Zeile an, dass der Befehl in der folgenden Zeile fortgesetzt wird. Zwischen dem letzten Parameter und dem Fortsetzungszeichen muss ein Leerzeichen stehen.

## Windows Basiskonfiguration

Warteschlangenmanager können in IBM MQ Explorer oder über die Eingabeaufforderung erstellt und gestartet werden.

Bei Verwendung der Eingabeaufforderung:

1. Geben Sie zur Erstellung des Warteschlangenmanagers folgenden Befehl ein:

```
crtmqm -u dlqname -q winnt
```

Dabei gilt:

**winnt**

ist der Name des Warteschlangenmanagers

**-q**

gibt an, dass dieser der Standardwarteschlangenmanager sein soll

**-u dlqname**

gibt den Namen der unzustellbaren Nachrichtenwarteschlange an.

Dieser Befehl erstellt einen Warteschlangenmanager und eine Gruppe von Standardobjekten.

2. Starten Sie den Warteschlangenmanager über den folgenden Befehl:

```
strmqm winnt
```

, wobei *winnt* der Name ist, den der Warteschlangenmanager bei seiner Erstellung erhalten hat.

### Windows **Kanalkonfiguration für Windows**

Beispielkonfiguration, die für den Windows-Warteschlangenmanager vorgenommen werden muss, um einen angegebenen Kanal zu implementieren.

In den folgenden Abschnitten werden die erforderlichen Konfigurationsschritte für den Windows-Warteschlangenmanager zur Implementierung des in „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Es ist jeweils der MQSC-Befehl angegeben. Entweder starten Sie **runmqsc** aus einer Eingabeaufforderung und geben die einzelnen Befehle nacheinander ein, oder Sie erstellen eine Befehlsdatei mit den Befehlen.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for Windows und IBM MQ for AIX bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Gruppe von Werten im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.



Tabelle 6. Konfigurationsbeispiele für IBM MQ for Windows			
	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<b>Definition für lokalen Knoten</b>			
A	Name des Warteschlangenmanagers		<b>WINNT</b>
B	Name der lokalen Warteschlange		<b>WINNT.LOCALQ</b>
 <b>AIX</b>  <b>AIX</b> <b>Verbindung zu IBM MQ for AIX</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für AIX“ auf Seite 11 angegebenen entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>
D	Name der fernen Warteschlange		<b>AIXREMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>
F	Name der Übertragungwarteschlange		<b>AIX</b>
G	Name des Senderkanals (SNA)		<b>WINNT.AIX.SNA</b>
H	Name des Senderkanals (TCP)		<b>WINNT.AIX.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>AIX.WINNT.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>AIX.WINNT.TCP</b>

Tabelle 6. Konfigurationsbeispiele für IBM MQ for Windows (Forts.)

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<p><b>Solaris</b>   <b>Solaris</b>   <b>Verbindung zu IBM MQ for Solaris</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 39 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>Solaris</b>
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>Solaris</b>
G	Name des Senderkanals (SNA)		<b>WINNT.SOLARIS.SNA</b>
H	Name des Senderkanals (TCP)		<b>WINNT.SOLARIS.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>SOLARIS.WINNT.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>SOLARIS.WINNT.TCP</b>
<p><b>Linux</b>   <b>Linux</b>   <b>Verbindung zu IBM MQ for Linux</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 34 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>LINUX</b>
G	Name des Senderkanals (SNA)		<b>WINNT.LINUX.SNA</b>
H	Name des Senderkanals (TCP)		<b>WINNT.LINUX.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>LINUX.WINNT.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>LINUX.WINNT.TCP</b>
<p><b>IBM i</b>   <b>IBM i</b>   <b>Verbindung zu IBM MQ for IBM i</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für IBM i“ auf Seite 27 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>
D	Name der fernen Warteschlange		<b>AS400.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AS400.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>AS400</b>
G	Name des Senderkanals (SNA)		<b>WINNT.AS400.SNA</b>
H	Name des Senderkanals (TCP)		<b>WINNT.AS400.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>AS400.WINNT.SNA</b>
J	Name des Empfängerkanals (TCP)	H	<b>AS400.WINNT.TCP</b>

Tabelle 6. Konfigurationsbeispiele für IBM MQ for Windows (Forts.)

	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<p><b>z/OS z/OS Verbindung zu IBM MQ for z/OS</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für z/OS“ auf Seite 51 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>MVS</b>
D	Name der fernen Warteschlange		<b>MVS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>MVS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>MVS</b>
G	Name des Senderkanals (SNA)		<b>WINNT.MVS.SNA</b>
H	Name des Senderkanals (TCP)		<b>WINNT.MVS.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>MVS.WINNT.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>MVS.WINNT.TCP</b>
<p><b>z/OS z/OS Verbindung zu IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Beispiel für die Konfiguration gemeinsamer Kanäle“ auf Seite 59 angegebenen entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>QSG</b>
D	Name der fernen Warteschlange		<b>QSG.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>QSG.SHAREDQ</b>
F	Name der Übertragungswarteschlange		<b>QSG</b>
G	Name des Senderkanals (SNA)		<b>WINNT.QSG.SNA</b>
H	Name des Senderkanals (TCP)		<b>WINNT.QSG.TCP</b>
I	Name des Empfängerkanals (SNA)	G	<b>QSG.WINNT.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>QSG.WINNT.TCP</b>

**Windows** Senderkanaldefinitionen unter Verwendung der SNA in IBM MQ for Windows  
Codebeispiel:

```

def ql (AIX) +                               F
  usage(xmitq) +
  replace

def qr (AIX.REMOTEQ) +                       D
  rname(AIX.LOCALQ) +                       E
  rqmname(AIX) +                             C
  xmitq(AIX) +                               F
  replace

def chl (WINNT.AIX.SNA) chltype(sdr) +       G
  trptype(lu62) +
  conname(AIXCPIC) +                         18
  xmitq(AIX) +                               F
  replace

```



## Windows Empfängerkanaldefinitionen unter Verwendung der SNA in IBM MQ for Windows

Codebeispiel:

```
def ql (WINNT.LOCALQ) replace B
def chl (AIX.WINNT.SNA) chltype(rcvr) + I
  trptype(lu62) +
  replace
```

## Windows Senderkanaldefinitionen über TCP/IP in IBM MQ for Windows

Codebeispiel:

```
def ql (AIX) + usage(xmitq) + replace F
def qr (AIX.REMOTEQ) + rname(AIX.LOCALQ) + rqmname(AIX) + xmitq(AIX) + replace D E C F
def chl (WINNT.AIX.TCP) chltype(sdr) + H
  trptype(tcp) +
  conname(remote_tcpip_hostname) +
  xmitq(AIX) + F
  replace
```

## Windows Empfängerkanaldefinitionen unter Verwendung von TCP in IBM MQ for Windows

Codebeispiel:

```
def ql (WINNT.LOCALQ) replace B
def chl (AIX.WINNT.TCP) chltype(rcvr) + J
  trptype(tcp) +
  replace
```

## Windows Automatischer Start

In IBM MQ for Windows haben Sie die Möglichkeit, den Start eines Warteschlangenmanagers und dessen Kanalinitiatoren, Kanälen, Empfangsprogrammen und Befehlsservern zu automatisieren.

Verwenden Sie das Snap-in "IBM MQ Services", um die Services für den Warteschlangenmanager zu definieren. Wenn Sie die Tests der Kommunikationskonfiguration erfolgreich abgeschlossen haben, setzen Sie die relevanten Services im Snap-in auf **automatisch**. Diese Datei kann beim Systemstart vom mitgelieferten IBM MQ-Service gelesen werden.

Weitere Informationen finden Sie im Abschnitt [IBM MQ verwalten](#).

## Windows Kanäle als Prozesse oder Threads ausführen






IBM MQ for Windows bietet die Flexibilität, sendende Kanäle als Windows-Prozesse oder Windows-Threads auszuführen. Dies wird im Parameter MCATYPE der Senderkanaldefinition festgelegt.

In den meisten Installationen werden die sendenden Kanäle als Threads ausgeführt, da auf diese Weise die zur Unterstützung vieler gleichzeitiger Kanalverbindungen benötigte Größe des virtuellen Speichers und Realspeicher reduziert werden kann. Für eine NetBIOS-Verbindung ist jedoch ein eigener Prozess für den sendenden Message Channel Agent erforderlich.

## z/OS IBM MQ-Beispielkonfiguration für z/OS

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zwischen IBM MQ for z/OS- und IBM MQ-Produkten auf anderen Plattformen.

Dieses Beispiel gilt auch für folgende andere Plattformen:

-  Windows
-  AIX
-  Solaris
-  Linux
-  IBM i
- VSE/ESA

Außerdem sind folgende Verbindungen möglich:

- z/OS bis z/OS
- z/OS zu MVS
- MVS zu MVS

Unter „IBM MQ-Beispielkonfiguration für alle Plattformen“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## **Verbindung aufbauen**

Zum Aufbau einer Verbindung müssen eine Reihe von Einstellungen konfiguriert werden.

### **Eine LU 6.2-Verbindung einrichten**

Aktuelle Informationen zum Konfigurieren von SNA über TCP/IP finden Sie in der folgenden IBM Online-dokumentation: [Communications Server for z/OS](#).

### **Eine TCP-Verbindung einrichten**

Damit die richtigen Parameter für die Steuerung verteilter Warteschlangen verwendet werden, modifizieren Sie das Warteschlangenmanagerobjekt durch folgenden Befehl. Sie müssen dem Warteschlangenmanagerattribut TCPNAME den Namen des TCP-Adressraums hinzufügen.

```
ALTER QMGR TCPNAME(TCPIP)
```

Die TCP-Verbindung ist nun eingerichtet. Sie können die Konfiguration abschließen.

## **IBM MQ for z/OS-Konfiguration**

Die folgenden Schritte dienen zur Konfiguration von IBM MQ sowie zum Starten und Konfigurieren von Kanälen und Empfangsprogrammen.

1. Starten Sie den Kanalinitiator durch folgenden Befehl:

```
/cpf START CHINIT 1
```

2. Starten Sie ein LU 6.2-Empfangsprogramm mithilfe des folgenden Befehls:

```
/cpf START LSTR LUNAME( M1 ) TRPTYPE(LU62)
```

LUNAME von M1 bezieht sich auf den symbolischen Namen, den Sie Ihrer logischen Einheit (5) gegeben haben. Sie müssen TRPTYPE(LU62) angeben, andernfalls geht das Empfangsprogramm davon aus, dass Sie TCP verwenden möchten.

3. Starten Sie ein TCP-Empfangsprogramm mithilfe des folgenden Befehls:

```
/cpf START LSTR
```

Wenn Sie einen anderen Anschluss als 1414 (Standardanschluss in IBM MQ) nutzen möchten, geben Sie folgenden Befehl ein:

```
/cpf START LSTR PORT( 1555 )
```

IBM MQ-Kanäle werden nicht erfolgreich initialisiert, wenn bei der Kanalfestlegung erkannt wird, dass die Nachrichtenfolgennummern an den beiden Enden unterschiedlich sind. Unter Umständen müssen Sie diese Kanäle manuell zurücksetzen.

### **Kanalkonfiguration für z/OS**

Zur Implementierung der Beispielkanäle sind einige Konfigurationsschritte auf dem z/OS-Warteschlangenmanager erforderlich.

In den folgenden Abschnitten werden die erforderlichen Konfigurationsschritte für den z/OS-Warteschlangenmanager zur Implementierung des in „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for z/OS und IBM MQ for Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Werte im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.





ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<b>Definition für lokalen Knoten</b>			
A	Name des Warteschlangenmanagers		<b>MVS</b>
B	Name der lokalen Warteschlange		<b>MVS.LOCALQ</b>
<b>  <b>Verbindung zu IBM MQ for Windows</b></b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für Windows</a> “ auf Seite 46 verwendeten Werten wie angegeben entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>WINNT</b>
G	Name des Senderkanals (LU 6.2)		<b>MVS.WINNT.SNA</b>
H	Name des Senderkanals (TCP)		<b>MVS.WINNT.TCP</b>
I	Name des Empfängerkanals (LU 6.2)	G	<b>WINNT.MVS.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>WINNT.MVS.TCP</b>
<b>  <b>Verbindung zu IBM MQ for AIX</b></b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „ <a href="#">Kanalkonfiguration für AIX</a> “ auf Seite 11 verwendeten Werten wie angegeben entsprechen.			

Tabelle 7. Konfigurationsbeispiele für IBM MQ for z/OS (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
C	Name des fernen Warteschlangenmanagers	A	<b>AIX</b>
D	Name der fernen Warteschlange		<b>AIXREMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>AIX</b>
G	Name des Senderkanals (LU 6.2)		<b>MVS.AIX.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>MVS.AIX.TCP</b>
I	Name des Empfängerkanals (LU 6.2)	G	<b>AIX.MVS.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>AIX.MVS.TCP</b>
<p><b>Solaris</b>   <b>Solaris</b>   <b>Verbindung zu IBM MQ for Solaris</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Solaris“ auf Seite 39 verwendeten Werten wie angegeben entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>Solaris</b>
D	Name der fernen Warteschlange		<b>SOLARIS.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>SOLARIS.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>Solaris</b>
G	Name des Senderkanals (LU 6.2)		<b>MVS.SOLARIS.SNA</b>
H	Name des Senderkanals (TCP)		<b>MVS.SOLARIS.TCP</b>
I	Name des Empfängerkanals (LU 6.2)	G	<b>SOLARIS.MVS.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>SOLARIS.MVS.TCP</b>
<p><b>Linux</b>   <b>Linux</b>   <b>Verbindung zu IBM MQ for Linux</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Linux“ auf Seite 34 verwendeten Werten wie angegeben entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>LINUX</b>
D	Name der fernen Warteschlange		<b>LINUX.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>LINUX.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>LINUX</b>
G	Name des Senderkanals (LU 6.2)		<b>MVS.LINUX.SNA</b>
H	Name des Senderkanals (TCP)		<b>MVS.LINUX.TCP</b>
I	Name des Empfängerkanals (LU 6.2)	G	<b>LINUX.MVS.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>LINUX.MVS.TCP</b>
<p><b>IBM i</b>   <b>IBM i</b>   <b>Verbindung zu IBM MQ for IBM i</b></p> <p>Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für IBM i“ auf Seite 27 verwendeten Werten wie angegeben entsprechen.</p>			
C	Name des fernen Warteschlangenmanagers	A	<b>AS400</b>

Tabelle 7. Konfigurationsbeispiele für IBM MQ for z/OS (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
D	Name der fernen Warteschlange		AS400.REMOTEQ
E	Name der Warteschlange auf fernem System	B	AS400.LOCALQ
F	Name der Übertragungswarteschlange		AS400
G	Name des Senderkanals (LU 6.2)		MVS.AS400.SNA
H	Name des Senderkanals (TCP/IP)		MVS.AS400.TCP
I	Name des Empfängerkanals (LU 6.2)	G	AS400.MVS.SNA
J	Name des Empfängerkanals (TCP/IP)	H	AS400.MVS.TCP

### z/OS **Senderkanaldefinitionen von IBM MQ for z/OS**

In diesem Abschnitt werden die zur Konfiguration von IBM MQ for z/OS unter Verwendung von LU 6.2 oder TCP erforderlichen Senderkanaldefinitionen ausführlich beschrieben.

Für LU 6.2:

```

Local Queue
  Object type : QLOCAL
                Name : WINNT           F
                Usage : X (XmitQ)

Remote Queue
  Object type : QREMOTE
                Name : WINNT.REMOTEQ   D
Name on remote system : WINNT.LOCALQ   E
Remote system name : WINNT             C
Transmission queue : WINNT             F

Sender Channel
  Channel name : MVS.WINNT.SNA         G
  Transport type : L (LU6.2)
Transmission queue name : WINNT         F
Connection name : M3                   13
    
```

Für TCP:

```

Local Queue
  Object type : QLOCAL
                Name : WINNT           F
                Usage : X (XmitQ)

Remote Queue
  Object type : QREMOTE
                Name : WINNT.REMOTEQ   D
Name on remote system : WINNT.LOCALQ   E
Remote system name : WINNT             C
Transmission queue : WINNT             F

Sender Channel
  Channel name : MVS.WINNT.TCP         H
  Transport type : T (TCP)
Transmission queue name : WINNT         F
Connection name : winnt.tcpip.hostname
    
```

### z/OS **Empfängerkanaldefinitionen von IBM MQ for z/OS**

In diesem Abschnitt werden die zur Konfiguration von IBM MQ for z/OS unter Verwendung von LU 6.2 oder TCP erforderlichen Empfängerkanaldefinitionen ausführlich beschrieben.

Für LU 6.2:

```

Local Queue
  Object type : QLOCAL
             Name : MVS.LOCALQ      B
             Usage : N (Normal)

Receiver Channel
  Channel name : WINNT.MVS.SNA      I
  
```

Für TCP:

```

Local Queue
  Object type : QLOCAL
             Name : MVS.LOCALQ      B
             Usage : N (Normal)

Receiver Channel
  Channel name : WINNT.MVS.TCP      J
  
```

## **IBM MQ-Beispielkonfiguration für z/OS unter Verwendung von QSGs**

Dieser Abschnitt enthält ein Beispiel für die Einrichtung von Kommunikationsverbindungen zu einer Gruppe mit gemeinsamer Warteschlange von IBM MQ unter Windows und AIX. Sie können auch Verbindungen von z/OS zu z/OS herstellen.

Die Einrichtung von Kommunikationsverbindungen zwischen einer Gruppe mit gemeinsamer Warteschlange und einer anderen Plattform als z/OS ist mit dem Vorgang identisch, der unter „[IBM MQ-Beispielkonfiguration für z/OS](#)“ auf Seite 49 beschrieben wird. Dieser Abschnitt enthält auch Beispiele anderer Plattformen.

Sobald die Verbindung eingerichtet ist, müssen Sie einige Kanäle definieren, um die Konfiguration abzuschließen. Dieser Vorgang wird im Abschnitt „[Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS](#)“ auf Seite 59 beschrieben.

Unter „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 finden Sie Hintergrundinformationen zu diesem Abschnitt und dessen Verwendung.

## **Konfigurationsparameter für eine LU 6.2-Verbindung**

Im folgenden Arbeitsblatt sind alle Parameter aufgelistet, die zum Einrichten der Kommunikation von einem z/OS-System zu einer der anderen IBM MQ-Plattformen erforderlich sind. Das Arbeitsblatt enthält Beispiele der Parameter, die in einer Arbeitsumgebung getestet wurden, und ermöglicht Ihnen, Ihre eigenen Werte einzutragen.

Die erforderlichen Schritte zum Einrichten einer LU 6.2 -Verbindung werden in „[LU 6.2-Verbindung in einer Gruppe mit gemeinsamer Warteschlange aufbauen](#)“ auf Seite 56 mit nummerierten Querverweisen auf die Parameter im Beispiel beschrieben.

Zahlen in der Referenzspalte geben an, dass der Wert mit dem Wert im entsprechenden Beispiel an anderer Stelle in diesem Abschnitt übereinstimmen muss. Die nachfolgenden Beispiele in diesem Abschnitt beziehen sich auf die Werte in der Spalte mit der ID. Die Einträge in der Spalte mit dem Parameternamen werden in „[Begriffserklärungen](#)“ auf Seite 55 erläutert.

Tabelle 8. Konfigurationsbeispiele für z/OS mit LU 6.2			
ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<b>Definition für lokalen Knoten mithilfe generischer Ressourcen</b>			
1	Befehlspräfix		/cpf

Tabelle 8. Konfigurationsbeispiele für z/OS mit LU 6.2 (Forts.)

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
2	Netz-ID		<b>NETID</b>
3	Knotenname		<b>MVSPU</b>
6	Modename		<b>#INTER</b>
7	Name des lokalen Transaktionsprogramms		<b>MQSERIES</b>
8	LAN-Zieladresse		<b>400074511092</b>
9	Name der lokalen logischen Einheit		<b>MVSLU1</b>
10	Name der generischen Ressource		<b>MVSGR</b>
11	Symbolisches Ziel		<b>G1</b>
12	Symbolisches Ziel für den Namen der generischen Ressource		<b>G2</b>
<b>Windows Windows Verbindung mit einem Windows-System</b>			
13	Symbolisches Ziel		<b>M3</b>
14	Modename	21	<b>#INTER</b>
15	Name des fernen Transaktionsprogramms	7	<b>MQSERIES</b>
16	Name der Partner-LU	5	<b>WINNTLU</b>
21	ID des fernen Knotens	4	<b>05D 30F65</b>
<b>AIX AIX Verbindung mit einem AIX-System</b>			
13	Symbolisches Ziel		<b>M4</b>
14	Modename	18	<b>#INTER</b>
15	Name des fernen Transaktionsprogramms	6	<b>MQSERIES</b>
16	Name der Partner-LU	4	<b>AIX-LU</b>

### **z/OS Begriffserklärungen**

Eine Erläuterung der im Arbeitsblatt zur Konfiguration verwendeten Begriffe.

#### **1 Befehlspräfix**

Dieser Begriff bezeichnet das eindeutige Befehlspräfix für das Warteschlangenmanager-Subsystem Ihres IBM MQ for z/OS-Systems. Der z/OS-Systemprogrammierer definiert diesen Wert bei der Installation in SYS1.PARMLIB(IEFSSNs); bei ihm können Sie den Wert erfragen.

#### **2 Netz-ID**

Die VTAM-Startprozedur in Ihrer Installation wird teilweise vom Member ATCSTRxx des Datensatzes angepasst, auf das DDNAME VTAMLST verweist. Die Netz-ID ist der für den Parameter NETID in diesem Member angegebene Wert. Für die Netz-ID müssen Sie den Namen des Parameters NETID angeben, der Eigner des IBM MQ-Kommunikationssubsystems ist. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

#### **3 Knotenname**

VTAM verfügt als Low Entry-Netzknoten nicht über einen Steuerpunktnamen für Advanced Peer-to-Peer Networking (APPN). Ein Steuerpunktnamen für Systemservice (SSCPNAME) liegt jedoch vor. Für den Knotennamen müssen Sie den Namen des Parameters SSCP angeben, der Eigner des IBM MQ-Kommunikationssubsystems ist. Dieser Wert ist im gleichen Member ATCSTRxx als Netz-ID angegeben. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

## 9 Name der lokalen LU

Eine logische Einheit (logical unit, LU) ist eine Software, die als Schnittstelle bzw. als Übersetzer zwischen einem Transaktionsprogramm und dem Netz dient. Sie verwaltet den Austausch von Daten zwischen Transaktionsprogrammen. Der Name der lokalen logischen Einheit ist die eindeutige Anwendungs-ID der VTAM dieses IBM MQ-Subsystems. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

## 11 12 13 Symbolisches Ziel

Dieser Begriff steht für den Namen, den Sie dem Nebeninformationsprofil CPI-C gegeben haben. Für jedes LU 6.2-Empfangsprogramm muss ein Nebeninformationseintrag vorhanden sein.

## 6 14 Modusname

Dies ist der festgelegte Name der Parametergruppe, die den LU 6.2-Datenaustausch steuert. Ein Eintrag mit diesem Namen und ähnliche Attribute müssen an beiden Enden der Sitzung definiert werden. In VTAM entspricht dies einem Modustableneintrag. Ihr Netzadministrator kann Ihnen diesen Tabelleneintrag zuweisen.

## 7 15 Transaktionsprogrammname

IBM MQ-Anwendungen, die versuchen, Daten mit diesem Warteschlangenmanager auszutauschen, geben einen symbolischen Namen für das Programm an, das am Empfangsende ausgeführt werden soll. Dieses ist im Attribut TPNAME der Kanaldefinition beim Sender festgelegt. Verwenden Sie der Einfachheit halber wenn möglich immer den Transaktionsprogrammnamen MQSERIES und bei Verbindungen mit VSE/ESA, wenn die Länge des Namens auf 4 Byte beschränkt ist, verwenden Sie MQTP.

Weitere Informationen finden Sie im Abschnitt LU6.2-Verbindung für z/OS mit APPC/MVS definieren.

## 8 LAN-Zieladresse

Dies steht für die LAN-Zieladresse, die Ihre Partnerknoten für die Kommunikation mit diesem Host verwenden. Wenn Sie mit einem 3745-Netzcontroller arbeiten, ist dies der im Parameter LOCADD angegebene Wert für die Leitungsdefinition, mit der Ihr Partner physisch verbunden ist. Wenn Ihre Partnerknoten andere Geräte verwenden, beispielsweise 317X- oder 6611-Geräte, wird die Adresse bei der Anpassung dieser Geräte festgelegt. Den Wert können Sie bei Ihrem Netzadministrator erfragen.

## 10 Name der generischen Ressource

Ein generischer Ressourcenname ist ein eindeutiger Name, der einer Gruppe von LU-Namen zugewiesen ist, die von den Kanalinitiatoren in einer Gruppe mit gemeinsamer Warteschlange verwendet werden.

## 16 Name der Partner-LU

Dies ist der LU-Name des IBM MQ-Warteschlangenmanagers in dem System, mit dem Sie die Kommunikation einrichten. Dieser Wert ist im Nebeninformationseintrag für den fernen Partner angegeben.

## 21 ID des fernen Knotens

Für eine Verbindung zu Windows ist diese ID die ID des lokalen Knotens in dem Windows-System, mit dem Sie die Kommunikation einrichten.

## LU 6.2-Verbindung in einer Gruppe mit gemeinsamer Warteschlange aufbauen

Eine LU 6.2-Verbindung wird in zwei Schritten hergestellt. Sie definieren sich selbst im Netz und Sie definieren eine Verbindung zum Partner.

### **Selbstdefinition im Netz mithilfe generischer Ressourcen**

Sie können VTAM Generic Resources verwenden, um einen einzigen Verbindungsnamen für die Verbindung zur Gruppe mit gemeinsamer Warteschlange zu erhalten.

1. SYS1.PARMLIB(APPCPMxx) enthält die Startparameter für APPC. Dieser Datei müssen Sie eine Zeile hinzufügen, um APPC mitzuteilen, wo sich die Nebeninformationen befinden. Die Zeile muss folgendes Format aufweisen:



```
SIDEINFO
DATASET (APPC . APPCSI)
```

2. Fügen Sie SYS1.PARMLIB(APPCPMxx) eine weitere Zeile hinzu, um den Namen der lokalen LU zu definieren, der für das Empfangsprogramm der IBM MQ LU 6.2-Gruppe verwendet werden soll. Diese Zeile muss folgendes Format haben:

```
LUADD ACBNAME(mvslu1)
      NOSCHED
      TPDATA(csq.appctp)
      GRNAME(mvsgtr)
```

Geben Sie Werte für ACBNAME (9), TPDATA und GRNAME (10) an.

Der Parameter NOSCHED teilt APPC mit, dass die neue logische Einheit nicht den LU 6.2-Scheduler (ASCH) verwendet, jedoch über keinen eigenen verfügt. TPDATA bezieht sich auf den Transaktionsprogramm-Datensatz, in dem LU 6.2 Informationen über Transaktionsprogramme speichert. Auch diesen Parameter verwendet IBM MQ nicht, er ist jedoch aufgrund der Syntax des Befehls LUADD erforderlich.

3. Starten Sie das APPC-Subsystem mit dem Befehl:

```
START APPC, SUB=MSTR, APPC=xx
```

Dabei ist *xx* das Suffix des PARMLIB-Members, in dem Sie die LU in Schritt 1 hinzugefügt haben.

**Anmerkung:** Wenn die APPC bereits ausgeführt wird, kann Sie durch folgenden Befehl aktualisiert werden:

```
SET APPC=xx
```

Die Auswirkungen hiervon sind kumulativ, das heißt, APPC vergisst keine Informationen zu Objekten, die bereits in diesem Member oder einem anderen PARMLIB-Member definiert sind.

4. Fügen Sie die neue LU einer geeigneten VTAM-Hauptknotendefinition hinzu. Diese befinden Sie in der Regel in SYS1.VTAMLST. Die APPL-Definition sieht wie das abgebildete Beispiel aus.

```

MVSLU APPL ACBNAME=MVSLU1,      9
          APPXC=YES,
          AUTOSES=0,
          DDRAINL=NALLOW,
          DLOGMOD=#INTER,      6
          DMINWML=10,
          DMINWNR=10,
          DRESPL=NALLOW,
          DSESLIM=60,
          LMDENT=19,
          MODETAB=MTCICS,
          PARSESS=YES,
          VERIFY=NONE,
          SECACPT=ALREADYV,
          SRBEXIT=YES
```

5. Aktivieren Sie den Hauptknoten. Diese Aktivierung kann durch folgenden Befehl erfolgen:

```
V, NET, ACT, majornode
```

6. Fügen Sie der CPI-C-Nebeninformationsdatei Einträge hinzu, die Ihre LU und den generischer Ressourcennamen definieren. Verwenden Sie hierfür das APPC-Dienstprogramm ATBSDFMU. Die Beispiel-JCL befindet sich in *thlqual.SCSQPROC(CSQ4SIDE)* (hierbei steht *thlqual* für das übergeordnete Qualifikationsmerkmal für die Zielbibliothek für IBM MQ-Datensätze in der Installation).

Die hinzuzufügenden Einträge haben das im folgenden Beispiel dargestellte Format:

```

SIADD
  DESTNAME(G1)          11
  MODENAME(#INTER)
  TPNAME(MQSERIES)
  PARTNER_LU(MVSLU1)    9
SIADD
  DESTNAME(G2)          12
  MODENAME(#INTER)
  TPNAME(MQSERIES)
  PARTNER_LU(MVSGR)     10

```

7. Damit die richtigen Parameter für die Steuerung verteilter Warteschlangen verwendet werden, modifizieren Sie das Warteschlangenmanagerobjekt durch folgenden Befehl. Sie müssen die lokale LU (9) angeben, die Ihrem Warteschlangenmanager im Attribut LUGROUP des Warteschlangenmanagers zugewiesen ist.

```
ALTER QMGR LUGROUP(MVSLU1)
```

### **Verbindung zu einem Partner definieren**

Sie können eine Verbindung zu einem Partner definieren, indem Sie der CPI-C-Nebeninformationsdatei einen Eintrag hinzufügen.

**Anmerkung:** Dieses Beispiel gilt für eine Verbindung zu einem Windows-System, die Aufgabe ist jedoch für andere Plattformen die gleiche.

Fügen Sie zur Definition der Verbindung in der CPI-C-Nebeninformationsdatei einen Eintrag hinzu. Eine Beispiel-JCL für diese Definition befindet sich in *thlqual.SCSQPROC(CSQ4SIDE)*.

Der hinzuzufügende Eintrag sieht wie folgt aus:

```

SIADD
  DESTNAME(M3)          13
  MODENAME(#INTER)     14
  TPNAME(MQSERIES)     15
  PARTNER_LU(WINNTLU)  16

```

### **Wie geht es weiter?**

Die Verbindung ist jetzt hergestellt. Sie können die Konfiguration abschließen.

Fahren Sie dazu mit „Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS“ auf Seite 59 fort.

### **TCP-Verbindung über Sysplex Distributor aufbauen**

Sie können Sysplex Distributor so einrichten, dass ein einziger Verbindungsname für die Verbindung zur Gruppe mit gemeinsamer Warteschlange verwendet wird.

1. Definieren Sie durch folgende Schritte eine verteilte DVIPA-Adresse:
  - a. Fügen Sie IPCONFIG eine DYNAMICXCF-Anweisung hinzu. Diese Anweisung dient zur Verbindung zwischen Images mithilfe dynamisch erstellter XCF TCP/IP-Verknüpfungen.
  - b. Verwenden Sie den Block VIPADYNAMIC auf allen Images im Sysplex.
    - i) Fügen Sie auf dem Eigner-Image eine VIPADEFINE-Anweisung in den Code ein, um die DVIPA zu erstellen. Fügen Sie anschließend eine VIPADISTRIBUTE-Anweisung in den Code ein, um sie auf alle anderen oder ausgewählte Images zu verteilen.
    - ii) Fügen Sie auf dem Backup-Image eine VIPABACKUP-Anweisung für die DVIPA-Adresse ein.
2. Wenn mehrere Kanalinitiatoren auf einer beliebigen logischen Partition im Sysplex gestartet werden, fügen Sie für den gemeinsamen Anschluss in die Reservierungsliste PORT im Datensatz PROFILE die Option SHAREPORT ein.

Weitere Informationen finden Sie unter [PORT statement](#) im Handbuch *z/OS Communications Server: IP Configuration Reference*.

Sysplex Distributor verteilt die Eingangsverbindungen gleichmäßig zwischen den einzelnen logischen Partitionen. Wenn sich auf einer logischen Partition mehrere Kanalinitiatoren befinden, wird durch die Verwendung von SHAREPORT die Eingangsverbindung an den Anschluss des Empfangsprogramms mit der geringsten Anzahl an Verbindungen übergeben.

Nachdem diese Schritte ausgeführt wurden, ist die TCP-Verbindung aufgebaut. Sie können die Konfiguration abschließen.

Wechseln Sie zu „[Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS](#)“ auf Seite 59.

## **Konfiguration von gemeinsamen Kanälen in IBM MQ for z/OS**

Konfigurieren Sie den gemeinsamen Kanal, indem Sie den Kanalinitiator starten und die entsprechenden Befehle für Ihre Konfiguration ausgeben.

1. Starten Sie den Kanalinitiator durch folgenden Befehl:

```
/cpf START CHINIT
```

2. Starten Sie durch Eingabe des folgenden Befehls ein Empfangsprogramm für die LU 6.2-Gruppe:

```
/cpf START LSTR TRPTYPE(LU62) LUNAME( G1 ) INDISP(GROUP)
```

LUNAME von G1 bezieht sich auf den symbolischen Namen, den Sie Ihrer logischen Einheit (11) gegeben haben.

3. Wenn Sie in Verbindung mit Sysplex Distributor mit virtueller IP-Adressvergabe arbeiten und über eine bestimmte Adresse empfangsbereit sein möchten, verwenden Sie folgenden Befehl:

```
/cpf START LSTR TRPTYPE(TCP) PORT(1555) IPADDR( musvipa ) INDISP(GROUP)
```

Es kann immer nur eine Instanz des gemeinsam genutzten Kanals aktiv sein. Versuche, eine zweite Instanz des Kanals zu starten, schlagen fehl (die ausgegebene Fehlermeldung variiert und hängt von weiteren Faktoren ab). Die gemeinsame Synchronisationswarteschlange verfolgt den Kanalstatus.

IBM MQ-Kanäle werden nicht erfolgreich initialisiert, wenn bei der Kanalfestlegung erkannt wird, dass die Nachrichtenfolgennummern an den beiden Enden unterschiedlich sind. Unter Umständen müssen Sie diese manuell zurücksetzen.

## **Beispiel für die Konfiguration gemeinsamer Kanäle**

Zur Konfiguration eines gemeinsamen Kanals sind eine Reihe von Schritten erforderlich.

In den folgenden Abschnitten werden die erforderlichen Konfigurationsschritte für den z/OS-Warteschlangenmanager zur Implementierung des in „[IBM MQ-Beispielkonfiguration für alle Plattformen](#)“ auf Seite 5 beschriebenen Kanals ausführlich erläutert.

Die Beispiele werden für eine Verbindung zwischen IBM MQ for z/OS und Windows bereitgestellt. Um eine Verbindung zu IBM MQ auf einer anderen Plattform herzustellen, geben Sie die jeweilige Gruppe von Werten aus der Tabelle anstelle der Werte im Beispiel für Windows ein.

**Anmerkung:** Die Wörter in **Fettschrift** sind empfohlene Werte und spiegeln die Namen der IBM MQ -Objekte wider, die in diesen Beispielen verwendet werden. Sie können sie in Ihrer Produktinstallation ändern, aber wenn Sie dies tun, stellen Sie sicher, dass Sie Ihre eigenen Werte verwenden, wenn Sie die Beispiele in diesem Abschnitt durcharbeiten.

Tabelle 9. Konfigurationsbeispiele für IBM MQ for z/OS bei Verwendung von Gruppen mit gemeinsamer Warteschlange

ID	Parameter Name (Parametername)	Referenz	Verwendetes Beispiel
<b>Definition für lokalen Knoten</b>			
A	Name des Warteschlangenmanagers		<b>QSG</b>
B	Name der lokalen Warteschlange		<b>QSG.SHAREDQ</b>
<div style="background-color: #800000; color: white; padding: 2px; display: inline-block;">Windows</div> <div style="background-color: #800000; color: white; padding: 2px; display: inline-block; margin-left: 10px;">Windows</div> <b>Verbindung zu IBM MQ for Windows</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für Windows“ auf Seite 46 verwendeten Werten wie angegeben entsprechen.			
C	Name des fernen Warteschlangenmanagers	A	<b>WINNT</b>
D	Name der fernen Warteschlange		<b>WINNT.REMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>WINNT.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>WINNT</b>
G	Name des Senderkanals (LU 6.2)		<b>QSG.WINNT.SNA</b>
H	Name des Senderkanals (TCP)		<b>QSG.WINNT.TCP</b>
I	Name des Empfängerkanals (LU 6.2)	G	<b>WINNT.QSG.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>WINNT.QSG.TCP</b>
<div style="background-color: #008000; color: white; padding: 2px; display: inline-block;">AIX</div> <div style="background-color: #008000; color: white; padding: 2px; display: inline-block; margin-left: 10px;">AIX</div> <b>Verbindung zu IBM MQ for AIX</b>			
Die Werte in diesem Abschnitt der Tabelle müssen den in „Kanalkonfiguration für AIX“ auf Seite 11 verwendeten Werten wie angegeben entsprechen.			
C	Name des fernen Warteschlangenmanagers		<b>AIX</b>
D	Name der fernen Warteschlange		<b>AIXREMOTEQ</b>
E	Name der Warteschlange auf fernem System	B	<b>AIX.LOCALQ</b>
F	Name der Übertragungswarteschlange		<b>AIX</b>
G	Name des Senderkanals (LU 6.2)		<b>QSG.AIX.SNA</b>
H	Name des Senderkanals (TCP/IP)		<b>QSG.AIX.TCP</b>
I	Name des Empfängerkanals (LU 6.2)	G	<b>AIX.QSG.SNA</b>
J	Name des Empfängerkanals (TCP/IP)	H	<b>AIX.QSG.TCP</b>

### z/OS Definitionen gemeinsamer Senderkanäle in IBM MQ for z/OS

Eine Beispielformatdefinition für gemeinsame Senderkanäle für LU 6.2 und TCP.

#### LU 6.2 verwenden

```

Local Queue
  Object type : QLOCAL
  Name       : WINNT           F
  Usage     : X (XmitQ)
  Disposition : SHARED

Remote Queue
  Object type : QREMOTE
  Name       : WINNT.REMOTEQ   D
  
```

```

Name on remote system : WINNT.LOCALQ      E
Remote system name   : WINNT              C
Transmission queue   : WINNT              F
Disposition          : GROUP

Sender Channel
Channel name         : MVS.WINNT.SNA      G
Transport type       : L (LU6.2)
Transmission queue name : WINNT          F
Connection name      : M3                 13
Disposition          : GROUP

```

## TCP verwenden

```

Local Queue
Object type : QLOCAL
Name        : WINNT              F
Usage       : X (XmitQ)
Disposition : SHARED

Remote Queue
Object type : QREMOTE
Name        : WINNT.REMOTEQ      D
Name on remote system : WINNT.LOCALQ E
Remote system name   : WINNT      C
Transmission queue   : WINNT      F
Disposition          : GROUP

Sender Channel
Channel name         : QSG.WINNT.TCP    H
Transport type       : T (TCP)
Transmission queue name : WINNT          F
Connection name      : winnt.tcpip.hostname
Disposition          : GROUP

```

## Definitionen gemeinsamer Empfängerkanäle in IBM MQ for z/OS

Eine Beispielformatdefinition gemeinsamer Empfängerkanäle für LU 6.2 und TCP.

## LU 6.2 verwenden

```

Local Queue
Object type : QLOCAL
Name        : QSG.SHAREDQ        B
Usage       : N (Normal)
Disposition : SHARED

Receiver Channel
Channel name : WINNT.QSG.SNA      I
Disposition  : GROUP

```

## TCP verwenden

```

Local Queue
Object type : QLOCAL
Name        : QSG.SHAREDQ        B
Usage       : N (Normal)
Disposition : SHARED

Receiver Channel
Channel name : WINNT.QSG.TCP      J
Disposition  : GROUP

```

## z/OS MQ-Beispielkonfiguration für z/OS unter Verwendung der gruppeninternen Warteschlangensteuerung

In diesem Abschnitt wird beschrieben, wie eine typische Anwendung zur Lohnbuchhaltungsabfrage, die derzeit zur Übertragung kurzer Nachrichten zwischen Warteschlangenmanagern die verteilte Steuerung von Warteschlangen verwendet, für die Nutzung von Gruppen mit gemeinsamen Warteschlangen und gemeinsam genutzten Warteschlangen migriert werden kann.

Zur Veranschaulichung der Verwendung der verteilten Steuerung von Warteschlangen, gruppeninternen Warteschlangensteuerung mit gemeinsam genutzten Warteschlangen und gemeinsam genutzten Warteschlangen werden drei Konfigurationen beschrieben. Die zugehörigen Diagramme zeigen nur den Datenfluss in eine Richtung, und zwar vom Warteschlangenmanager QMG1 zum Warteschlangenmanager QMG3.

### z/OS Konfiguration 1

Konfiguration 1 beschreibt, wie derzeit mit der verteilten Steuerung von Warteschlangen Nachrichten zwischen den Warteschlangenmanagern QMG1 und QMG3 übertragen werden.

Konfiguration 1 zeigt ein System für die verteilte Warteschlangensteuerung, mit dem Nachrichten, die Warteschlangenmanager QMG1 von der Lohnbuchhaltungsabfrage empfängt, an Warteschlangenmanager QMG2 und schließlich an Warteschlangenmanager QMG3 übertragen werden, von dem aus sie dann an den Lohnbuchhaltungsserver gesendet werden.

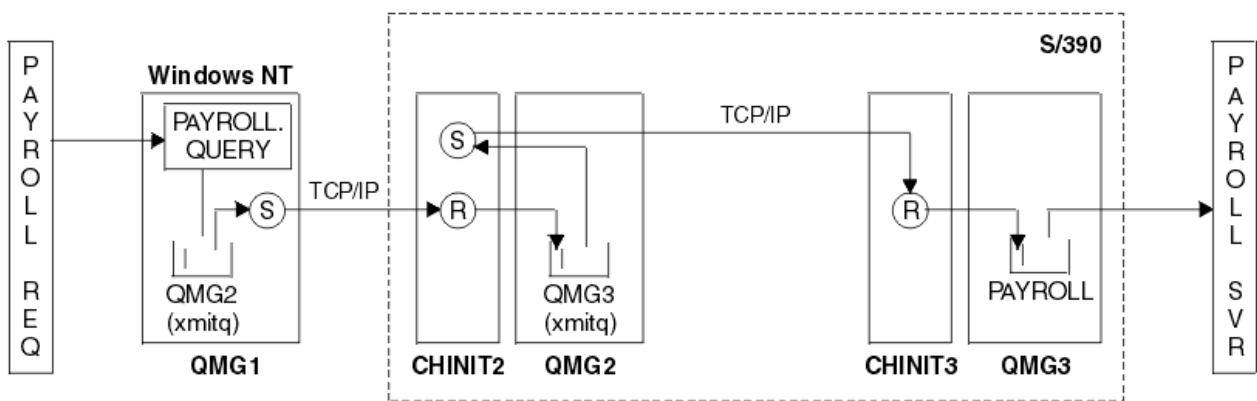


Abbildung 2. Konfiguration 1: z/OS unter Verwendung der gruppeninternen Warteschlangensteuerung

Der Ablauf der Operationen lautet wie folgt:

1. Eine Abfrage wird unter Verwendung der Anwendung zur Lohnbuchhaltungsabfrage eingegeben, die mit dem Warteschlangenmanager QMG1 verbunden ist.
2. Die Anwendung zur Lohnbuchhaltungsabfrage reiht die Abfrage in die ferne Warteschlange PAYROLL.QUERY ein. Die Warteschlange PAYROLL.QUERY wird in die Übertragungswarteschlange QMG2 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG2 eingereicht.
3. Der Senderkanal (S) auf Warteschlangenmanager QMG1 übergibt die Abfrage an den Partnerempfängerkanal (R) auf Warteschlangenmanager QMG2.
4. Der Empfängerkanal (R) auf Warteschlangenmanager QMG2 reiht die Abfrage in die Warteschlange PAYROLL auf Warteschlangenmanager QMG3 ein. Die Warteschlange PAYROLL auf QMG3 wird in die Übertragungswarteschlange QMG3 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG3 eingereicht.
5. Der Senderkanal (S) auf Warteschlangenmanager QMG2 übergibt die Abfrage an den Partnerempfängerkanal (R) auf Warteschlangenmanager QMG3.
6. Der Empfängerkanal (R) auf Warteschlangenmanager QMG3 reiht die Abfrage in die lokale Warteschlange PAYROLL ein.

7. Die Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist, ruft die Abfrage aus der lokalen Warteschlange PAYROLL ab, verarbeitet sie und generiert eine passende Antwort.

## **Definitionen für Konfiguration 1**

Konfiguration 1 erfordert die nachfolgenden Definitionen (beachten Sie, dass die Definitionen kein Auslösen berücksichtigen und dass nur Kanaldefinitionen zur Kommunikation mittels TCP/IP angegeben werden).

### **Auf QMG1**

Definition der fernen Warteschlange:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +  
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG2')
```

Definition der Empfangswarteschlange für Antworten:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

### **Auf QMG2**

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)  
  
DEFINE QLOCAL(QMG3) DESCR('Transmission queue to QMG3') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

Hier ersetzen Sie WINTQMG1(1414) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG3') XMITQ(QMG3) CONNAME('MVSQMG3(1416)')
```

Hier ersetzen Sie MVSQMG3(1416) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG1')
```

```
DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG3')
```

## Auf QMG3

Definition der lokalen Warteschlange:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +  
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE
```

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG3.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG3) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG2')
```

## Konfiguration 2

Konfiguration 2 beschreibt, wie Nachrichten mittels Gruppen mit gemeinsamer Warteschlange und der gruppeninternen Warteschlangensteuerung ohne Auswirkung auf die Lohnbuchhaltungsserveranwendung auf dem Back-End zwischen den Warteschlangenmanagern QMG1 und QMG3 übertragen werden können.

Konfiguration 2 zeigt ein System für die verteilte Warteschlangensteuerung, mit dem Nachrichten unter Verwendung von Gruppen mit gemeinsamer Warteschlange und gruppeninterner Warteschlangensteuerung von der Anwendung zur Lohnbuchhaltungsabfrage an den Lohnbuchhaltungsserver übertragen werden. Dank dieser Konfiguration entfallen Kanaldefinitionen zwischen den Warteschlangenmanagern QMG2 und QMG3, da die Nachrichten mittels der gruppeninternen Warteschlangensteuerung zwischen diesen beiden Warteschlangenmanagern übertragen werden.



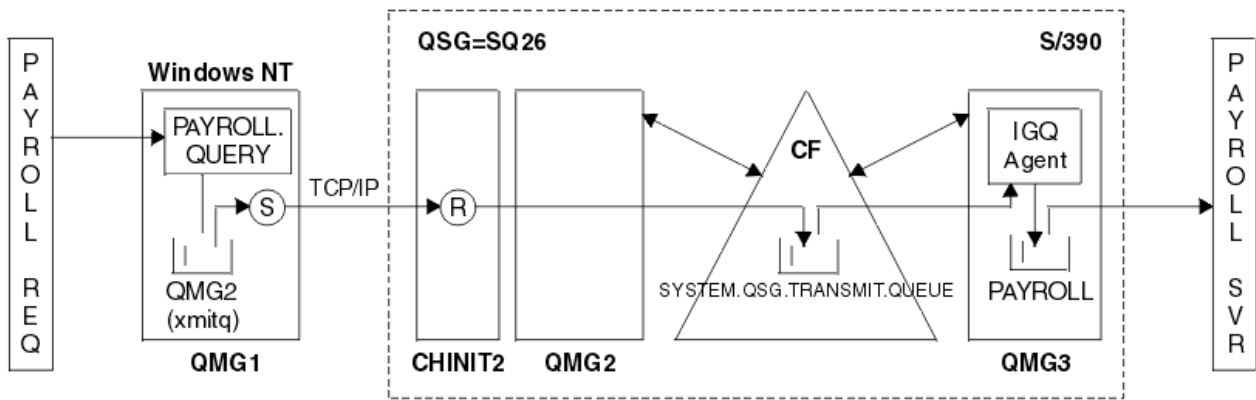


Abbildung 3. Konfiguration 2

Der Ablauf der Operationen lautet wie folgt:

1. Eine Abfrage wird unter Verwendung der Anwendung zu Lohnbuchhaltungsabfrage eingegeben, die mit dem Warteschlangenmanager QMG1 verbunden ist.
2. Die Anwendung zur Lohnbuchhaltungsabfrage reiht die Abfrage in die ferne Warteschlange PAYROLL.QUERY ein. Die Warteschlange PAYROLL.QUERY wird in die Übertragungswarteschlange QMG2 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG2 eingereiht.
3. Der Senderkanal (S) auf Warteschlangenmanager QMG1 übergibt die Abfrage an den Partnerempfängerkanal (R) auf Warteschlangenmanager QMG2.
4. Der Empfängerkanal (R) auf Warteschlangenmanager QMG2 reiht die Abfrage in die Warteschlange PAYROLL auf Warteschlangenmanager QMG3 ein. Die Warteschlange PAYROLL auf QMG3 wird in die gemeinsame Übertragungswarteschlange YSTEM.QSG.TRANSMIT.QUEUE aufgelöst und die Abfrage wird in die gemeinsame Übertragungswarteschlange SYSYSTEM.QSG.TRANSMIT.QUEUE eingereiht.
5. Der IGQ-Agent auf dem Warteschlangenmanager QMG3 ruft die Abfrage aus der gemeinsamen Übertragungswarteschlange SYSTEM.QSG.TRANSMIT.QUEUE ab und stellt sie in die lokale Warteschlange PAYROLL auf dem Warteschlangenmanager QMG3.
6. Die Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist, ruft die Abfrage aus der lokalen Warteschlange PAYROLL ab, verarbeitet sie und generiert eine passende Antwort.

**Anmerkung:** Im Beispiel zur Lohnbuchhaltungsabfrage werden nur kompakte Nachrichten übertragen. Wenn Sie sowohl persistente als auch nicht persistente Nachrichten übertragen müssen, kann eine Kombination aus Konfiguration 1 und Konfiguration 2 eingerichtet werden. Auf diese Weise lassen sich lange Nachrichten unter Verwendung der Route der verteilten Warteschlangensteuerung übertragen, während kompakte Nachrichten über die potenziell schnellere Route der Warteschlangensteuerung übertragen werden.

## **z/OS** Definitionen für Konfiguration 2

Konfiguration 2 erfordert die nachfolgenden Definitionen (beachten Sie, dass die Definitionen kein Auslösen berücksichtigen und dass nur Kanaldefinitionen zur Kommunikation mittels TCP/IP angegeben werden).

Es wird angenommen, dass die Warteschlangenmanager QMG2 und QMG3 bereits als Mitglieder derselben Gruppe mit gemeinsamer Warteschlange konfiguriert wurden.

### **Auf QMG1**

Definition der fernen Warteschlange:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG2')
```

Definition der Empfangswarteschlange für Antworten:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

## Auf QMG2

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)  
  
DEFINE QLOCAL(SYSTEM.QSG.TRANSMIT.QUEUE) QSGDISP(SHARED) +  
DESCR('IGQ Transmission queue') REPLACE PUT(ENABLED) USAGE(XMITQ) +  
GET(ENABLED) INDXTYPE(CORRELID) CFSTRUCT('APPLICATION1') +  
DEFSOPT(SHARED) DEFPSIST(NO)
```

Hier ersetzen Sie APPLICATION1 mit dem Namen Ihrer definierten CF-Struktur. Beachten Sie auch, dass diese Warteschlange, da es eine gemeinsam genutzte Warteschlange ist, nur auf einem der Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange definiert werden muss.

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) REPLACE +  
DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

Hier ersetzen Sie WINTQMG1(1414) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG1')
```

Warteschlangenmanagerdefinition:

```
ALTER QMGR IGQ(ENABLED)
```

## Auf QMG3

Definition der lokalen Warteschlange:

```
DEFINE QLOCAL(PAYROLL) DESCR('Payroll query request queue') REPLACE +  
PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE
```

Warteschlangenmanagerdefinition:

```
ALTER QMGR IGQ(ENABLED)
```

### Konfiguration 3

Konfiguration 3 beschreibt, wie Nachrichten mittels Gruppen mit gemeinsamer Warteschlange und gemeinsam genutzten Warteschlangen ohne Auswirkung auf die Lohnbuchhaltungsserveranwendung auf dem Back-End zwischen den Warteschlangenmanagern QMG1 und QMG3 übertragen werden können.

Konfiguration 3 zeigt ein System für die verteilte Warteschlangensteuerung, mit dem Nachrichten unter Verwendung von Gruppen mit gemeinsamer Warteschlange und gemeinsam genutzten Warteschlangen zwischen den Warteschlangenmanagern QMG1 und QMG3 ausgetauscht werden.

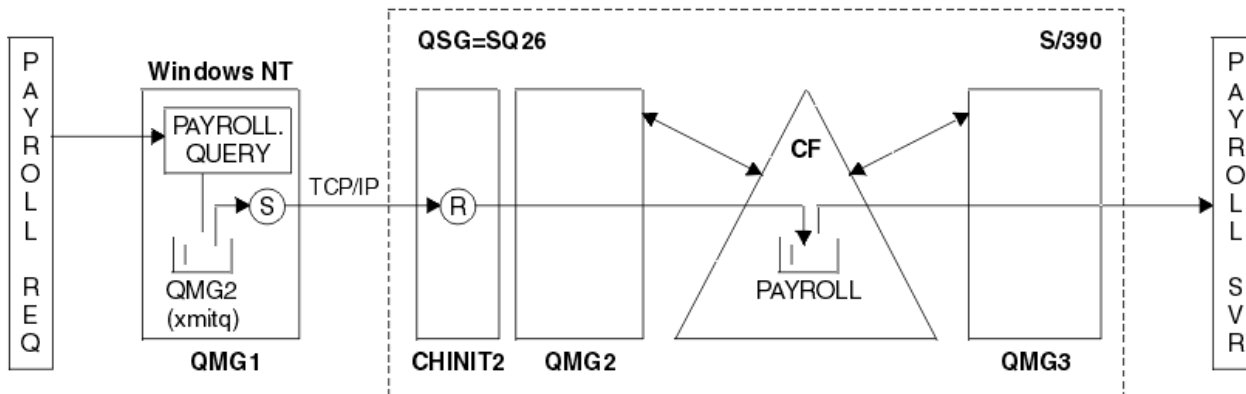


Abbildung 4. Konfiguration 3

Der Verfahrensablauf gliedert sich wie folgt:

1. Eine Abfrage wird unter Verwendung der Anwendung zur Lohnbuchhaltungsabfrage eingegeben, die mit dem Warteschlangenmanager QMG1 verbunden ist.
2. Die Anwendung zur Lohnbuchhaltungsabfrage reißt die Abfrage in die ferne Warteschlange PAYROLL.QUERY ein. Die Warteschlange PAYROLL.QUERY wird in die Übertragungswarteschlange QMG2 aufgelöst und die Abfrage wird in die Übertragungswarteschlange QMG2 eingereiht.
3. Der Senderkanal (S) auf Warteschlangenmanager QMG1 übergibt die Abfrage an den Partnerempfängerkanal (R) auf Warteschlangenmanager QMG2.
4. Der Empfängerkanal (R) auf Warteschlangenmanager QMG2 reißt die Abfrage in die gemeinsam genutzte Warteschlange PAYROLL ein.
5. Die Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist, ruft die Abfrage aus der gemeinsam genutzten Warteschlange PAYROLL ab, verarbeitet sie und generiert eine passende Antwort.

Diese Konfiguration ist zweifellos am einfachsten zu konfigurieren. Um Antworten (generiert von der Lohnbuchhaltungsserveranwendung, die mit Warteschlangenmanager QMG3 verbunden ist) von Warteschlangenmanager QMG3 an Warteschlangenmanager QMG2 und anschließend an Warteschlangenmanager QMG1 zu übertragen, muss jedoch eine verteilte Steuerung von Warteschlangen oder eine gruppeninterne Warteschlangensteuerung konfiguriert werden. (Informationen zur Konfiguration, die beim

Rücksenden von Antworten an die Anwendung zur Lohnbuchhaltungsabfrage verwendet wird, finden Sie unter „Erläuterung des Beispiels der Gruppe mit gemeinsamer Warteschlange für z/OS“ auf Seite 175.)

Auf QMG3 sind keine Definitionen erforderlich.

### **Definitionen für Konfiguration 3**

Konfiguration 3 erfordert die nachfolgenden Definitionen (beachten Sie, dass die Definitionen kein Auslösen berücksichtigen und dass nur Kanaldefinitionen zur Kommunikation mittels TCP/IP angegeben werden).

Es wird angenommen, dass die Warteschlangenmanager QMG2 und QMG3 bereits als Mitglieder derselben Gruppe mit gemeinsamer Warteschlange konfiguriert wurden.

#### **Auf QMG1**

Definition der fernen Warteschlange:

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QMG3') REPLACE +  
PUT(ENABLED) RNAME(PAYROLL) RQMNAME(QMG3) XMITQ(QMG2)
```

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG2) DESCR('Transmission queue to QMG2') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QMG2') XMITQ(QMG2) CONNAME('MVSQMG2(1415)')
```

Hier ersetzen Sie MVSQMG2(1415) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG2')
```

Definition der Empfangswarteschlange für Antworten:

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to payroll queries sent to QMG3')
```

#### **Auf QMG2**

Definition der Übertragungswarteschlange:

```
DEFINE QLOCAL(QMG1) DESCR('Transmission queue to QMG1') REPLACE +  
PUT(ENABLED) USAGE(XMITQ) GET(ENABLED)
```

Senderkanaldefinitionen (für TCP/IP):

```
DEFINE CHANNEL(QMG2.TO.QMG1) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QMG1') XMITQ(QMG1) CONNAME('WINTQMG1(1414)')
```

Hier ersetzen Sie WINTQMG1(1414) durch den Namen und Port Ihrer Warteschlangenmanagerverbindung.

Empfängerkanaldefinition (für TCP/IP):

```
DEFINE CHANNEL(QMG1.TO.QMG2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QMG1')
```

Definition der lokalen Warteschlange:

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) DESCR('Payroll query request queue') +  
REPLACE PUT(ENABLED) USAGE(NORMAL) GET(ENABLED) SHARE +  
DEFSOPT(SHARED) DEFPSIST(NO) CFSTRUCT(APPLICATION1)
```

Hier ersetzen Sie APPLICATION1 mit dem Namen Ihrer definierten CF-Struktur. Beachten Sie auch, dass diese Warteschlange, da es eine gemeinsam genutzte Warteschlange ist, nur auf einem der Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange definiert werden muss.

## Auf QMG3

Auf QMG3 sind keine Definitionen erforderlich.

### Beispiel ausführen

Nachdem Sie das Beispiel eingerichtet haben, können Sie es ausführen.

Für Konfiguration 1:

1. Starten Sie die Warteschlangenmanager QMG1, QMG2 und QMG3.
2. Starten Sie die Kanalinitiatoren für QMG2 und QMG3.
3. Starten Sie die Empfangsprogramme auf QMG1, um an Port 1414 empfangsbereit zu sein, auf QMG2, um an Port 1415 empfangsbereit zu sein und auf QMG3, um an Port 1416 empfangsbereit zu sein.
4. Starten Sie die Senderkanäle auf QMG1, QMG2 und QMG3.
5. Starten Sie die mit QMG1 verbundene Anwendung zur Lohnbuchhaltungsabfrage.
6. Starten Sie die mit QMG3 verbundene Lohnbuchhaltungsserveranwendung.
7. Übergeben Sie eine Lohnbuchhaltungsabfrage an QMG3 und warten Sie auf die Antwort der Lohnbuchhaltung.

Für Konfiguration 2:

1. Starten Sie die Warteschlangenmanager QMG1, QMG2 und QMG3.
2. Starten Sie den Kanalinitiator für QMG2.
3. Starten Sie die Empfangsprogramme auf QMG1, um an Port 1414 empfangsbereit zu sein, und auf QMG2, um an Port 1415 empfangsbereit zu sein.
4. Starten Sie den Senderkanal auf QMG1 und QMG2.
5. Starten Sie die mit QMG1 verbundene Anwendung zur Lohnbuchhaltungsabfrage.
6. Starten Sie die mit QMG3 verbundene Lohnbuchhaltungsserveranwendung.
7. Übergeben Sie eine Lohnbuchhaltungsabfrage an QMG3 und warten Sie auf die Antwort der Lohnbuchhaltung.

Für Konfiguration 3:

1. Starten Sie die Warteschlangenmanager QMG1, QMG2 und QMG3.
2. Starten Sie den Kanalinitiator für QMG2.
3. Starten Sie die Empfangsprogramme auf QMG1, um an Port 1414 empfangsbereit zu sein, und auf QMG2, um an Port 1415 empfangsbereit zu sein.
4. Starten Sie die Senderkanäle auf QMG1 und QMG2.
5. Starten Sie die mit QMG1 verbundene Anwendung zur Lohnbuchhaltungsabfrage.

6. Starten Sie die mit QMG3 verbundene Lohnbuchhaltungsserveranwendung.
7. Übergeben Sie eine Lohnbuchhaltungsabfrage an QMG3 und warten Sie auf die Antwort der Lohnbuchhaltung.

## **Beispiel erweitern**

Das Beispiel kann auf zahlreiche Art erweitert werden.

Sie können das Beispiel:

- Zur Verwendung von Kanal-Triggering und Anwendungs-Triggering (Warteschlange PAYROLL und PAYROLL.REPLY) erweitern
- Zur Kommunikation unter Verwendung von LU6.2 konfigurieren
- Zur Konfiguration von weiteren Warteschlangenmanagern für die Gruppe für gemeinsame Warteschlange erweitern. Danach können Sie die Serveranwendung klonen, sodass sie auf anderen Warteschlangenmanagerinstanzen ausgeführt werden kann, um der Abfragewarteschlange PAYROLL mehrere Server bereitzustellen.
- Zur Erhöhung der Instanzen der Anwendung zur Lohnbuchhaltungsabfrage erweitern, um die Verarbeitung von Abfragen aus mehreren Clients zu veranschaulichen
- Zur Verwendung von Sicherheit (IGQAUT und IGQUSER) erweitern

## **Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen**

Nachfolgend ist die Sicherheit beschrieben, die auf die Dateien und Verzeichnisse unter `/var/mqm/` angewendet wird; außerdem wird erläutert, weshalb die Dateisystemberechtigungen auf die jeweiligen Werte gesetzt sind. Um sicherzustellen, dass IBM MQ ordnungsgemäß funktioniert, sollten Sie die von IBM MQ festgelegten Dateisystemberechtigungen nicht ändern.

### **crtmqdir-Befehl**

Wenn Ihr Unternehmen eine der `/var/mqm`-Dateiberechtigungen geändert hat, können Sie die Berechtigungen aktualisieren oder Verzeichnisse hinzufügen, indem Sie den Befehl **`crtmqdir`** verwenden.

### **IBM MQ-Dateisystemsicherheit unter UNIX, Linux und IBM i**

In den Dateien unter dem IBM MQ-Datenverzeichnis (`/var/mqm`) wird Folgendes gespeichert:

- IBM MQ-Konfigurationsdaten
- Anwendungsdaten (IBM MQ-Objekte und die in IBM MQ-Nachrichten enthaltenen Daten)
- Steuerinformationen der Laufzeit
- Überwachungsdaten (Nachrichten und FFST-Dateien)

Der Zugriff auf diese Daten wird über Dateisystemberechtigungen gesteuert, wobei einige der Daten für alle Benutzer zugänglich sind, während andere Daten auf Mitglieder der IBM MQAdministratorgruppe `mqm` (oder `QMQM` unter IBM i) beschränkt sind.

Die Erteilung des Zugriffs erfolgt in den folgenden drei Kategorien:

#### **Nur Gruppe 'mqm'**

Auf die Dateien und Verzeichnisse in dieser Kategorie können nur IBM MQ-Administratoren (Mitglieder der Gruppe `mqm`) und die IBM MQ-Warteschlangenmanagerprozesse zugreifen.

Die Dateiberechtigungen für diese Dateien und Verzeichnisse lauten wie folgt:

```
-IWXIXW--- mqm:mqm      (UNIX and Linux)
-IWXIXW--- QMQMADM:QMQM (IBM i)
```

Es folgt ein Beispiel der Dateien und Verzeichnisse in dieser Kategorie:

```
/var/mqm/qmgrs/QMGR/qm.ini
/var/mqm/qmgrs/QMGR/channel/
/var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
/var/mqm/qmgrs/QMGR/queues/
/var/mqm/qmgrs/QMGR/queues/SYSTEM!DEFAULT!LOCAL!QUEUES/
/var/mqm/qmgrs/QMGR/errors/
/var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
/var/mqm/qmgrs/QMGR/ssl/
/var/mqm/qmgrs/QMGR/@qmgr/
/var/mqm/qmgrs/QMGR/@qmpersist/
...
```

## Lesezugriff für alle Benutzer - Lese- und Schreibzugriff für Mitglieder der Gruppe 'mqm'

Die Dateien und Verzeichnisse in dieser Kategorie können von allen Benutzern gelesen werden, aber nur Mitglieder der Gruppe mqm können diese Dateien ändern und diese Verzeichnisse bearbeiten.

Die Dateiberechtigungen für diese Dateien und Verzeichnisse lauten wie folgt:

```
-IWXIXWXI-X mqm:mqm      (UNIX and Linux)
-IWXIXWXI-X QMQMADM:QMQM (IBM i)
```

Es folgt ein Beispiel der Dateien und Verzeichnisse in dieser Kategorie:

```
/var/mqm/mqs.ini
/var/mqm/exits/
/var/mqm/qmgrs/
/var/mqm/qmgrs/QMGR/
/var/mqm/qmgrs/QMGR/@app/
/var/mqm/qmgrs/QMGR/@ipcc/
```



**Achtung:** Sie sollten Ausführungsberechtigungen nur für ausführbare Dateien und Scripts festlegen. Beispiel: Unter Linux werden bei der Ausführung des Befehls **crtmqm** die folgenden Dateiberechtigungen festgelegt:

```
-rW-rW---- mqm mqm /var/mqm/qmgrs/QMGR/qm.ini
-rW-rW---- mqm mqm /var/mqm/qmgrs/QMGR/channel/SYSTEM!DEF!SCRVONN
-rW-rW---- mqm mqm /var/mqm/qmgrs/QMGR/errors/AMQERR01.LOG
-rW-rW-r-- mqm mqm /var/mqm/mqs.ini
```

IBM MQ 8.0:

```
/var/mqm/sockets/@SYSTEM
/var/mqm/sockets/QMGR/@app/hostname
/var/mqm/sockets/QMGR/@ipcc/hostname
```

## Lese- und Schreibzugriff für alle Benutzer Dateien mit Lese- und Schreibzugriff für alle Benutzer

IBM MQ hat keine *Regulär*-Dateien mit globalem Schreibzugriff (777). Es gibt jedoch eine Reihe von *Spezial*-Dateien, die global modifizierbare Dateiberechtigungen haben.

Diese speziellen Dateien bedeuten keine Sicherheitslücke. Obwohl die Berechtigungen als 777 angezeigt werden, handelt es sich nicht um *Regulär*-Dateien und Sie können nicht direkt in sie schreiben.

Es handelt sich hierbei um die folgenden speziellen Dateien:

## Symbolische Links

Symbolische Links werden durch das Zeichen `l` am Anfang ihrer Berechtigungen identifiziert. Die Berechtigungen für die symbolische Verbindung haben keinen Einfluss darauf, wer auf die Zieldatei zugreifen kann. Der Grund hierfür ist, dass der Zugriff auf den Befehl durch die Berechtigungen auf dem Ziel der symbolischen Verbindung gesteuert wird.

Da die Berechtigungen für symbolische Verbindungen auf den meisten UNIX- und Linux-Systemen nicht geändert werden können, werden sie stets als `l-rwxrwxrwx` angezeigt.

## Socketdateien

Socketdateien sind spezielle Dateien, die vom Betriebssystem als Folge eines Prozesses erstellt werden, der ein UNIX-Domänensocket erstellt. Diese Dateien können durchs am Anfang der Dateiberechtigungen identifiziert werden, d. h. `s-rwxrwxrwx`.

Die Berechtigungen für die Datei bedeuten nicht, dass der Zugriff auf die Datei selbst erteilt wird, sondern sie definieren, wer sich mit dem UNIX-Domänensocket verbinden kann.

IBM MQ verwendet mehrere dieser Socketdateien und die Festlegung der Berechtigungen richtet sich immer danach, wem die Kommunikation mit dem Socket erlaubt wird.

Die folgenden Verzeichnisse enthalten Socketdateien mit Lese-/Schreibberechtigungen für alle Benutzer (`s-rwxrwxrwx`).

IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketEC/hostname/Zsocket_*
```

Socketdateien, die von Anwendungen verwendet werden, welche sich unter Verwendung von isolierten Bindungen mit IBM MQ verbinden.

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/*
```

## Verzeichnisse mit Lese- und Schreibzugriff für alle Benutzer

In manchen Fällen müssen IBM MQ-Anwendungen Dateien unter dem IBM MQ-Datenverzeichnis erstellen. Um sicherzustellen, dass Anwendungen bei Bedarf Dateien erstellen können, wird für verschiedene Verzeichnisse ein globaler Schreibzugriff erteilt. Dies bedeutet, dass jeder Benutzer auf dem System Dateien in diesem Verzeichnis erstellen kann.

Mit Ausnahme der Fehlerprotokolldateien, in die von jedem Mitglied der Gruppe `mqm` geschrieben werden kann, werden alle in diesen Verzeichnissen erstellten Dateien mit eingeschränkten Berechtigungen erstellt, die nur den Schreibzugriff des Dateierstellers zulassen. Dadurch kann der Systemadministrator die Benutzer-ID aller Daten verfolgen, die in Dateien dieser Verzeichnisse geschrieben werden.

### **`/var/mqm/errors/`**

Dieses Verzeichnis enthält die Systemfehlerprotokolldateien und FFST-Dateien. Die Berechtigung für dieses Verzeichnis lautet `drwxrwsrwt`, d. h., alle Benutzer auf dem System können Dateien in diesem Verzeichnis erstellen.

Das `SetGroupId`-Bit zeigt an, dass alle Dateien, die in diesem Verzeichnis erstellt werden, das Gruppeneigentum `mqm` haben.

Das Sticky Bit `t` ist für dieses Verzeichnis standardmäßig nicht festgelegt, aber ein IBM MQ-Administrator kann dies explizit festlegen, damit Benutzer nur die Dateien löschen können, die sie erstellen.

**Anmerkung:**  Diese Funktion ist unter IBM i nicht verfügbar.

### **`AMQERR0*.LOG`**

Diese Fehlerprotokolldateien können nur direkt von Mitgliedern der `group` geschrieben werden, aber jeder Benutzer kann die in diese Dateien geschriebenen Nachrichten lesen (Berechtigung: `-r-w-rw-r--`).



### AMQnnnnn.\*.FDC

Diese Dateien enthalten FFST-Informationen, die geschrieben werden, wenn im Warteschlangenmanager oder in einer von einem Benutzer geschriebenen Anwendung ein Fehler auftritt. Diese Dateien werden mit den Berechtigungen `-rw-r-----` erstellt.


### `/var/mqm/trace/`

Wenn der IBM MQ-Trace aktiviert ist, werden Tracedateien in dieses Verzeichnis geschrieben. Der IBM MQ-Trace wird von allen Prozessen geschrieben, die einem Warteschlangenmanager mit aktiviertem Trace zugeordnet sind.

Die Berechtigungen für dieses Verzeichnis sind `drwxrwsrwt`, d. h., alle Benutzer auf dem System können Dateien in diesem Verzeichnis erstellen.

Das `SetGroupId`-Bit `sg` gibt an, dass alle Dateien, die in diesem Verzeichnis erstellt werden, das Gruppeneigentumsrecht von `'mqm'` haben.

Das Sticky Bit `t` ist für dieses Verzeichnis standardmäßig nicht festgelegt, aber ein IBM MQ-Administrator kann dies explizit festlegen, damit Benutzer nur die Dateien löschen können, die sie erstellen.

**Anmerkung:**  Diese Funktion ist unter IBM i nicht verfügbar.

### AMQnnnnn.\*.TRC

Diese Dateien enthalten die Tracedaten, die von jedem Prozess geschrieben werden, für den ein Trace erstellt wird, und die mit den Berechtigungen `-rw-r-----` erstellt werden.

Die Berechtigungen für dieses Verzeichnis sind `drwxrwsrwt` und die Berechtigungen der in diesem Verzeichnis erstellten Socketdateien sind `srwx-----`.

### IBM MQ 8.0:

```
/var/mqm/sockets/QMGR/zsocketapp/hostname/
```


Dieses Verzeichnis wird von Anwendungen verwendet, die über *isolierte* Bindungen eine Verbindung zum IBM MQ -Warteschlangenmanager herstellen. Während der Verbindungsverarbeitung wird von der verbindenden Anwendung in diesem Verzeichnis eine Socketdatei erstellt. Nachdem die Verbindung zu dem Warteschlangenmanager hergestellt wurde, wird die Socketdatei entfernt.

Die Berechtigungen für dieses Verzeichnis sind `drwxrwsrwt` und die Berechtigungen der in diesem Verzeichnis erstellten Socketdateien sind `srwx-----`.

Das `SetGroupId`-Bit `sf` für dieses Verzeichnis stellt sicher, dass alle Dateien, die in diesem Verzeichnis erstellt werden, das Gruppeneigentumsrecht von `'mqm'` haben.

Auf allen Plattformen außer IBM i ist für diese Verzeichnisse auch das Sticky Bit `t` gesetzt, das verhindert, dass ein Benutzer Dateien mit Ausnahme der Dateien, deren Eigner er ist, löschen kann. So können nicht berechtigte Benutzer keine Dateien löschen, die ihnen nicht gehören.

```
/var/mqm/sockets/QMGR/@ipcc/ssem/hostname/  
/var/mqm/sockets/QMGR/@app/ssem/hostname/
```

 Für Prozesse, die über *gemeinsam genutzte* Bindungen eine Verbindung zu IBM MQ herstellen, können UNIX -Domänensockets für die Synchronisation zwischen der Anwendung und dem Warteschlangenmanager verwendet werden. Wenn UNIX-Domänensockets verwendet werden, wird die zugehörige Socketdatei in diesen Verzeichnissen erstellt.

Die Berechtigungen für diese Verzeichnisse sind `drwxrwsrwt` und die Berechtigungen der in diesen Verzeichnissen erstellten Socketdateien sind `srwxrwxrwx`.

Das `SetGroupId`-Bit `sf` für diese Verzeichnisse stellt sicher, dass alle in diesen Verzeichnissen erstellten Dateien das Gruppeneigentumsrecht `mqm` haben.

Auf allen Plattformen außer IBM i ist für diese Verzeichnisse auch das Sticky Bit `t` festgelegt, das verhindert, dass ein Benutzer Dateien mit Ausnahme der Dateien, deren Eigner er ist, löscht. So können nicht berechtigte Benutzer keine Dateien löschen, die ihnen nicht gehören.

## HOME

Bei Verwendung einer nicht registrierten oder nicht installierten Version von IBM MQ wie dem wiederverteilbaren Client wird das Verzeichnis `#{HOME}/.mqm` erstellt.

Das Verzeichnis wird erstellt, damit IBM MQ zuverlässig auf seine Socketdateien zugreifen kann, indem ein Pfad verwendet wird, der innerhalb der **sun\_path**-Länge liegt. Wenn IBM MQ keine Daten in das Verzeichnis HOME schreiben kann, erhalten Sie eine Fehlermeldung.

## Verwendung von System V-IPC-Ressourcen durch IBM MQ

IBM MQ verwendet gemeinsam genutzten System V-Speicher und Semaphoren für die Kommunikation zwischen Prozessen. Die Gruppierung dieser Ressourcen richtet sich danach, wie sie in den einzelnen Gruppen mit der entsprechenden Eigentümerschaft und den jeweiligen Zugriffsberechtigungen verwendet werden.

Für die Prüfung, welche System V-IPC-Ressourcen auf einem System zu IBM MQ gehören, haben Sie folgende Möglichkeiten:

- Prüfen Sie die Eigentümerschaft.

Der Eigner von IBM MQ System V IPC-Ressourcen ist immer der Benutzer 'mqm' auf UNIX -Plattformen und Linux. Unter IBM i ist der Benutzer 'QMQM' der Eigner.

- Verwenden Sie ab IBM MQ 8.0 das Dienstprogramm `amqspdbg`.

Das Dienstprogramm `amqspdbg`, das mit IBM MQ geliefert wird, kann zum Anzeigen des gemeinsam genutzten Speichers und der Semaphor-IDs für einen bestimmten Warteschlangenmanager verwendet werden.

Sie müssen den Befehl einmalig für die Gruppe 'system' der von IBM MQ erstellten System V-Ressourcen ausgeben:

```
# amqspbg -z -I
```

Anschließend müssen Sie ihn vier Mal für jeden Warteschlangenmanager auf dem System ausgeben, um die vollständige Liste der von IBM MQ verwendeten System V-Ressourcen zu erhalten. In den folgenden Beispielen wird der Warteschlangenmanagername `QMGR1` angenommen:

```
# amqspdbg -i QMGR1 -I
# amqspdbg -q QMGR1 -I
# amqspdbg -p QMGR1 -I
# amqspdbg -a QMGR1 -I
```

Die Zugriffsberechtigungen für die von IBM MQ erstellten System V-Ressourcen werden so festgelegt, dass berechtigten Benutzern immer die jeweils richtige Zugriffsstufe erteilt wird. Eine Reihe von System V-IPC-Ressourcen, die von IBM MQ erstellt wurden, sind für alle Benutzer auf der Maschine zugänglich und haben die Berechtigungen `-rw-rw-rw-`.

Der Parameter **-g Anwendungsgruppe** im Befehl `crtmqm` kann verwendet werden, um den Zugriff auf einen Warteschlangenmanager auf die Zugehörigkeit zu einer bestimmten Betriebssystemgruppe zu beschränken. Die Verwendung der Funktionalität einer Gruppe mit eingeschränkten Rechten beschränkt die für die System V-IPC-Ressourcen erteilten Berechtigungen noch detaillierter.

Linux

UNIX

## IBM MQ-Dateiberechtigungen in /opt/mqm mit setuid für mqm

Die folgenden Informationen beziehen sich auf die Situation, dass das Sicherheitsteam einige der ausführbaren IBM MQ-Dateien in der Verzeichnisstruktur `$MQ_INSTALLATION_PATH` markiert hat, weil sie lokale Sicherheitsrichtlinien verletzen. Die Standardposition unter AIX ist `/usr/mqm`; bei den anderen UNIX-Betriebssystemen lautet sie `/opt/mqm`. Die Informationen in diesem Abschnitt gelten auch dann, wenn Sie IBM MQ in einem anderen als dem Standardverzeichnis installiert haben (z. B. in `/opt/mqm90`) oder es in Ihrer Umgebung mehrere Installationen gibt.

## Problemursache

Das Sicherheitsteam hat unter `$MQ_INSTALLATION_PATH` die folgenden Problembereiche erkannt:

1. Dateien im Verzeichnis `/opt/mqm/bin` sind `setuid` für den Eigner der Verzeichnisstruktur, in der sie sich befinden. Beispiel:

```
dr-xr-xr-x   mqm mqm  ${MQ_INSTALLATION_PATH}/bin
-r-sr-s---   mqm mqm  ${MQ_INSTALLATION_PATH}/bin/addmqinf
-r-sr-s---   mqm mqm  ${MQ_INSTALLATION_PATH}/bin/amqcrista
-r-sr-s---   mqm mqm  ${MQ_INSTALLATION_PATH}/bin/amqfcxba
...
```

2. Eigner praktisch aller Verzeichnisse und Dateien ist "mqm:mqm", außer der folgenden, deren Eigner der Root-Benutzer ist:

```
dr-xr-x---   root mqm  ${MQ_INSTALLATION_PATH}/bin/security
-r-sr-x---   root mqm  ${MQ_INSTALLATION_PATH}/bin/security/amqoamax
-r-sr-x---   root mqm  ${MQ_INSTALLATION_PATH}/bin/security/amqoampx
```

Eigner dieses Unterverzeichnisses muss der Root-Benutzer sein, da es sich um die ausführbaren Dateien handelt, die mit dem Betriebssystem interagieren, wenn der Benutzer über einen IBM MQ-Client ein Kennwort angibt. Das Kennwort wird vom IBM MQ-Warteschlangenmanager an das Betriebssystem übergeben, um zu bestätigen, ob das Kennwort gültig oder ungültig ist.

3. Der Benutzer ist nicht Eigner von Dateien im Verzeichnis `/opt/mqm/lib/iconv` (dieses Verzeichnis ist unter AIX nicht vorhanden). Beispiel:

```
dr-xr-xr-x   mqm mqm  ${MQ_INSTALLATION_PATH}/lib/iconv
-r--r--r--   bin bin  ${MQ_INSTALLATION_PATH}/lib/iconv/002501B5.tbl
-r--r--r--   bin bin  ${MQ_INSTALLATION_PATH}/lib/iconv/002501F4.tbl
-r--r--r--   bin bin  ${MQ_INSTALLATION_PATH}/lib/iconv/00250333.tbl
...
```

4. Das Fixpack-Wartungsverzeichnis auf RPM-basierten Linux-Systemen. Wenn Fixpacks installiert werden, werden die vorhandenen Dateien unter diesem Verzeichnis in einer Struktur gespeichert, die der im folgenden Beispiel ähnelt, mit der Ausnahme, dass in diesem Beispiel V. R die IBM MQ -Versions- und Releasenummer und die Unterverzeichnisse darstellt, die von den installierten Fixpacks abhängig sind:

```
drwx-----   root root  ${MQ_INSTALLATION_PATH}/maintenance
drwxr-xr-x   root root  ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.1
drwxr-xr-x   root root  ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.3
drwxr-xr-x   root root  ${MQ_INSTALLATION_PATH}/maintenance/V.R.0.4
...
```

## Lösung des Problems

Eine der Bedenken in Bezug auf `Setuid`-Programme auf UNIX-Systemen war, dass die Systemsicherheit durch die Manipulation von Umgebungsvariablen wie `LD*` (`LD_LIBRARY_PATH`, `LIBPATH` auf AIX und so weiter) gefährdet werden könnte. Dies ist kein Problem mehr, weil verschiedene UNIX-Betriebssysteme diese `LD*`-Umgebungsvariablen jetzt beim Laden von `setuid`-Programmen ignorieren.

1. Warum sind einige der IBM MQ-Programme `mqm-setuid` oder `mqm-setgid`?

In IBM MQ sind die Benutzer-ID "mqm" und alle IDs, die Teil der Gruppe "mqm" sind, die IBM MQ-Benutzer mit Verwaltungsaufgaben.

IBM MQ-Warteschlangenmanagerressourcen werden geschützt, indem eine Authentifizierung dieser Benutzer durchgeführt wird. Da die Warteschlangenmanagerprozesse diese Warteschlangenmanagerressourcen verwenden und ändern, benötigen die Warteschlangenmanagerprozesse die Berechtigung "mqm", um auf die Ressourcen zugreifen zu können. Deshalb werden IBM MQ-Warteschlangenmanagerunterstützungsprozesse so entwickelt, dass sie mit der effektiven Benutzer-ID "mqm" ausgeführt werden.

Damit auch Benutzer ohne Verwaltungsaufgaben auf IBM MQ-Objekte zugreifen können, stellt IBM MQ den Objektberechtigungsmanager (OAM) bereit, mit dem Berechtigungen erteilt und entzogen werden

können, falls dies für die Anwendung, die vom Benutzer ohne Verwaltungsaufgaben ausgeführt wird, notwendig ist.

Dank der Möglichkeit, unterschiedliche Authentifizierungsebenen für Benutzer einzurichten, und der Tatsache, dass **setuid**- und **setgid**-Programme LD\*-Variablen ignorieren, wird die Sicherheit des Systems in keinerlei Weise durch die Binär- und Bibliotheksdateien von IBM MQ beeinträchtigt.

2. Ist es möglich, die Berechtigungen zu ändern, um die Sicherheitsrichtlinien des Unternehmens zu erfüllen, ohne die IBM MQ-Funktionalität zu gefährden?

Sie dürfen die Berechtigungen und Eigentumsrechte von IBM MQ -Binärdateien und -Bibliotheken nicht ändern. Die IBM MQ-Funktionalität kann unter solchen Änderungen leiden, sodass Warteschlangenmanagerprozesse möglicherweise nicht auf einige der Ressourcen zugreifen können.

Die Berechtigungen und die Eigentümerschaften stellen keinesfalls eine Sicherheitsbedrohung für das System dar.

Linux-Festplattenlaufwerke/Platten, auf denen IBM MQ installiert ist oder sich IBM MQ-Daten befinden, dürfen nicht mit der Option `nosuid` angehängt werden. Diese Konfiguration kann die Funktionalität von IBM MQ beeinträchtigen.

Weitere Informationen finden Sie unter „[Auf /var/mqm angewendete IBM MQ-Dateisystemberechtigungen](#)“ auf Seite 70.

## Zugehörige Konzepte

[Dateisystem](#)

## Windows IBM MQ-Dateisystemberechtigungen unter Windows

In den folgenden Informationen wird die Sicherheit beschrieben, die auf die Dateien und Verzeichnisse unter Windows angewendet wird. Um sicherzustellen, dass IBM MQ ordnungsgemäß funktioniert, sollten Sie die von IBM MQ festgelegten Dateisystemberechtigungen nicht ändern.

### Datenverzeichnis

**Anmerkung:** Die Berechtigungen, die für das Stammelement dieses Verzeichnisses festgelegt sind, werden in der gesamten Verzeichnisstruktur nach unten übernommen.

Für die Verzeichnisse unter dem Datenverzeichnis (DATADIR) sind die folgenden Berechtigungen festgelegt, abgesehen von den Ausnahmen, die im folgenden Text ausführlich beschrieben werden.

#### Administrators

Uneingeschränkter Zugriff

#### Gruppe 'mqm'

Uneingeschränkter Zugriff

#### SYSTEM

Uneingeschränkter Zugriff

#### Jeder

Lese- und Ausführungszugriff

Die Ausnahmen sind:

#### DATADIR\errors

Alle uneingeschränkten Zugriff

#### DATADIR\trace

Alle uneingeschränkten Zugriff

#### DATADIR\log

##### Administrators

Uneingeschränkter Zugriff

**Gruppe 'mqm'**

Uneingeschränkter Zugriff

**SYSTEM**

Uneingeschränkter Zugriff

**Jeder**

Lesen

**DATADIR\log\****Administrators**

Uneingeschränkter Zugriff

**Gruppe 'mqm'**

Uneingeschränkter Zugriff

**SYSTEM**

Uneingeschränkter Zugriff

Kein Zugriff erteilt für 'Alle'.

Die Fehlerprotokolldateien AMQERR01.LOG usw. übernehmen ihre Sicherheitseinstellungen nicht aus ihrem Verzeichnis, sondern werden stattdessen auf Everyone: Full Control (Alle: uneingeschränkter Zugriff) gesetzt.

**Frühere Releases des Produkts**

In Releases des Produkts vor IBM MQ 8.0 befanden sich das Standardprogramm und die Standarddatenverzeichnisse an einer gemeinsamen Position.

In jeder Installation, die ursprünglich vor IBM MQ 8.0 installiert wurde. Die Daten- und Programmverzeichnisse, die an den Standardpositionen installiert und anschließend von dort aktualisiert wurden, bleiben an derselben Position (in C:\Program Files\IBM\WebSphere MQ).

Falls sich die Daten- und Programmverzeichnisse an einer gemeinsamen Position befinden, gelten die obigen Informationen nur für die Verzeichnisse, die zum Datenverzeichnis gehören, und nicht für die Verzeichnisse, die Teil des Programmverzeichnisses sind.

## Benennungseinschränkungen für Warteschlangen

Es gibt Einschränkungen bei der Länge von Warteschlangennamen. Einige Warteschlangennamen sind für Warteschlangen reserviert, die vom Warteschlangenmanager definiert werden.

**Einschränkungen bei der Länge von Namen**

Die Namen von Warteschlangen können bis zu 48 Zeichen lang sein.

**Reservierte Warteschlangennamen**

Namen, die mit "SYSTEM." beginnen, sind für Warteschlangen reserviert, die vom Warteschlangenmanager definiert werden. Mit dem Befehl **ALTER** oder **DEFINE REPLACE** können Sie diese Warteschlangendefinitionen an Ihre Installation anpassen. Folgende Namen sind für IBM MQ definiert:

<i>Tabelle 10. Namen und Beschreibungen für reservierte Warteschlangen</i>	
<b>Name der Warteschlange</b>	<b>Beschreibung</b>
SYSTEM.ADMIN.ACTIVITY.QUEUE	Warteschlange für Aktivitätsberichte
SYSTEM.ADMIN.CHANNEL.EVENT	Warteschlange für Kanalereignisse
SYSTEM.ADMIN.COMMAND.EVENT	Warteschlange für Befehlsereignisse
SYSTEM.ADMIN.COMMAND.QUEUE	Warteschlange, an die PCF-Befehlsnachrichten gesendet werden

Tabelle 10. Namen und Beschreibungen für reservierte Warteschlangen (Forts.)

<b>Name der Warteschlange</b>	<b>Beschreibung</b>
SYSTEM.ADMIN.CONFIG.EVENT	Warteschlange für Konfigurationsereignisse
SYSTEM.ADMIN.PERFM.EVENT	Warteschlange für Leistungsereignisse
SYSTEM.ADMIN.PUBSUB.EVENT	Ereigniswarteschlange für Publish/Subscribe-Ereignisse des Systems
SYSTEM.ADMIN.QMGR.EVENT	Warteschlange für Warteschlangenmanagerereignisse
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Warteschlange für Traceroute-Antwornachrichten
SYSTEM.AUTH.DATA.QUEUE	Warteschlange mit den Zugriffskontrolllisten für den Warteschlangenmanager (nicht für z/OS)
SYSTEM.CHANNEL.INITQ	Initialisierungswarteschlange für Kanäle
SYSTEM.CHANNEL.SYNCQ	Warteschlange mit den Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM MQ-Kanalauthentifizierungsdaten
SYSTEM.CICS.INITIATION.QUEUE	Warteschlange zum Auslösen (nicht unter z/OS)
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange zur Kommunikation von Repository-Änderungen zwischen Warteschlangenmanagern
SYSTEM.CLUSTER.HISTORY.QUEUE	Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange mit Informationen zum Repository
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Die Warteschlange zum Erstellen der einzelnen Übertragungswarteschlangen für den jeweiligen Clustersenderkanal
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungswarteschlange für alle von der Clusterunterstützung verwalteten Ziele
SYSTEM.COMMAND.INPUT	Warteschlange, an die Befehlsnachrichten unter z/OS gesendet werden
SYSTEM.COMMAND.REPLY.MODEL	Definition der Modellwarteschlange für Antworten auf Befehle (unter z/OS)
SYSTEM.DEAD.LETTER.QUEUE	Warteschlange für nicht zustellbare Nachrichten (nicht unter z/OS)
SYSTEM.DEFAULT.ALIAS.QUEUE	Standarddefinition der Aliaswarteschlange
SYSTEM.DEFAULT.INITIATION.QUEUE	Warteschlange für die Auslösung eines angegebenen Prozesses (nicht unter z/OS)
SYSTEM.DEFAULT.LOCAL.QUEUE	Standarddefinition der lokalen Warteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standarddefinition der Modellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Standarddefinition der fernen Warteschlange
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	Lokale Warteschlange mit einer persistenten Kopie der permanenten Subskriptionen des Warteschlangenmanagers
SYSTEM.HIERARCHY.STATE	Warteschlange mit Informationen zum Status der Beziehungen zwischen Warteschlangenmanagern in einer Publish/Subscribe-Hierarchie

Tabelle 10. Namen und Beschreibungen für reservierte Warteschlangen (Forts.)

Name der Warteschlange	Beschreibung
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen
SYSTEM.INTERNAL.REPLY.QUEUE	IBM MQ-interne Antwortwarteschlange (nicht unter z/OS)
SYSTEM.INTER.QMGR.CONTROL	Warteschlange in einer Publish/Subscribe-Hierarchie für den Empfang von Anforderungen von einem fernen Warteschlangenmanager zur Erstellung einer Proxy-Subskription
SYSTEM.INTER.QMGR.PUBS	Warteschlange in einer Publish/Subscribe-Hierarchie für den Empfang von Veröffentlichungen von einem fernen Warteschlangenmanager
SYSTEM.INTER.QMGR.FANREQ	Warteschlange in einer Publish/Subscribe-Hierarchie für die Verarbeitung von Anforderungen zur Erstellung einer Proxy-Subskription auf einem fernen Warteschlangenmanager
SYSTEM.MQEXPLORER.REPLY.MODEL	Definition der Modellwarteschlange für Antworten auf IBM MQ Explorer
SYSTEM.MQSC.REPLY.QUEUE	Modellwarteschlangendefinition für Antworten auf WebSphere MQ-Scriptbefehle (nicht unter z/OS)
SYSTEM.QSG.CHANNEL.SYNCQ	Gemeinsam genutzte lokale Warteschlange zum Speichern von Nachrichten mit den Synchronisationsdaten für gemeinsame Kanäle (nur unter z/OS)
SYSTEM.QSG.TRANSMIT.QUEUE	Gemeinsam genutzte lokale Warteschlange, die vom Agenten zur gruppeninternen Warteschlangensteuerung für die Übertragung von Nachrichten zwischen Warteschlangenmanagern in der Gruppe mit gemeinsamer Warteschlange verwendet wird (nur unter z/OS)
SYSTEM.RETAINED.PUB.QUEUE	Lokale Warteschlange mit Kopien aller ständigen Veröffentlichungen des Warteschlangenmanagers
SYSTEM.SELECTION.EVALUATION.QUEUE	IBM MQ-interne Warteschlange für Auswahlbewertungen (nicht unter z/OS)
SYSTEM.SELECTION.VALIDATION.QUEUE	IBM MQ-interne Warteschlange für Auswahlprüfungen (nicht unter z/OS)

## Benennungseinschränkungen für andere Objekte

Es gibt Einschränkungen bei der Länge von Objektnamen. Einige Objektnamen sind für Objekte reserviert, die vom Warteschlangenmanager definiert werden.

### Einschränkungen bei der Länge von Namen

Die Namen von Prozessen, Namenslisten, Clustern, Themen, Services und Authentifizierungsdatenobjekten können eine Länge von maximal 48 Zeichen haben.

Die Namen von Kanälen können eine Länge von maximal 20 Zeichen haben.

Die Namen von Speicherklassen können eine Länge von maximal 8 Zeichen haben.

Die Namen von CF-Strukturen können eine Länge von maximal 12 Zeichen haben.

## Reservierte Objektnamen

Namen, die mit SYSTEM beginnen, sind für vom Warteschlangenmanager definierte Objekte reserviert. Mit den Befehlen **ALTER** oder **DEFINE REPLACE** können Sie diese Objektdefinitionen an Ihre Installation anpassen. Folgende Namen sind für IBM MQ definiert:

<i>Tabelle 11. Namen und Beschreibungen für reservierte Objekte</i>	
<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.ADMIN.SVRCONN	Serververbindungskanal für die ferne Verwaltung eines Warteschlangenmanagers
SYSTEM.AUTO.RECEIVER	Standardempfängerkanal für automatische Definition (nur unter UNIX, Linux, and Windows)
SYSTEM.AUTO.SVRCONN	Standardserververbindungskanal für automatische Definition (nur Multiplatforms)
SYSTEM.BASE.TOPIC	Basisartikel zur ASPARENT-Auflösung. Wenn ein besonderes administratives Artikelobjekt keine übergeordneten administrativen Artikelobjekte hat, werden alle ASPARENT-Attribute von diesem Objekt übernommen.
SYSTEM.DEF.CLNTCONN	Standarddefinition für Clientverbindungskanal
SYSTEM.DEF.CLUSRCVR	Standarddefinition für Clusterempfängerkanal
SYSTEM.DEF.CLUSSDR	Standarddefinition für Clustersenderkanal
SYSTEM.DEF.RECEIVER	Standarddefinition für Empfängerkanal
SYSTEM.DEF.REQUESTER	Standarddefinition für Requesterkanal
SYSTEM.DEF.SENDER	Standarddefinition für Senderkanal
SYSTEM.DEF.SERVER	Standarddefinition für Serverkanal
SYSTEM.DEF.SVRCONN	Standarddefinition für Serververbindungskanal
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Definition eines Standardauthentifizierungsdatenobjekts zur Definition von Authentifizierungsdatenobjekten vom Typ CRLLDAP
SYSTEM.DEFAULT.AUTHINFO.OCSP	Definition eines Standardauthentifizierungsdatenobjekts zur Definition von Authentifizierungsdatenobjekten vom Typ OCSP
SYSTEM.DEFAULT.LISTENER.LU62	Standardmäßiges SNA-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.NETBIOS	Standardmäßiges NetBIOS-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.SPX	Standardmäßiges SPX-Empfangsprogramm (nur unter Windows)
SYSTEM.DEFAULT.LISTENER.TCP	Standardmäßiges TCP/IP-Empfangsprogramm (nur Multiplatforms)
SYSTEM.DEFAULT.NAMELIST	Standardnamenslistendefinition
SYSTEM.DEFAULT.PROCESS	Standarddefinition für Prozesse
SYSTEM.DEFAULT.SERVICE	Standardservice (nur Multiplatforms)
SYSTEM.DEFAULT.TOPIC	Standarddefinition für Prozesse



Tabelle 11. Namen und Beschreibungen für reservierte Objekte (Forts.)	
Objektname	Beschreibung
SYSTEM.QPUBSUB.QUEUE.NAMELIST	Eine Liste von Warteschlangen für das zu überwachende Queued Publish/Subscribe Interface
z/OS z/OS SYSTEMST	Standardmäßige Speicherklassendefinition (nur unter z/OS)

## Auflösung des Warteschlangennamens

In größeren Netzen bietet die Verwendung von Warteschlangenmanagern gegenüber anderen Kommunikationsformen einige Vorteile. Diese Vorteile ergeben sich aus der Namensauflösungsfunktion im verteilten Warteschlangenmanagement, die sicherstellt, dass Warteschlangennamensauflösungen von Warteschlangenmanagern auf der Sende- und der Empfangsseite eines Kanals durchgeführt werden.

Die Hauptvorteile dieses Ansatzes sind:

- Anwendungen müssen keine Routing-Entscheidungen treffen.
- Anwendungen müssen die Netzstruktur nicht kennen.
- Netzverknüpfungen werden von Systemadministratoren erstellt.
- Die Netzstruktur wird von Netzplanern gesteuert.
- Zur Partitionierung des Datenverkehrs können mehrere Kanäle verwendet werden.

In der folgenden Abbildung wird ein Beispiel für die Auflösung von Warteschlangennamen gezeigt. Zu sehen sind zwei Maschinen in einem Netz; auf einer Maschine ist eine PUT-Anwendung aktiv, auf der anderen Maschine eine GET-Anwendung. Die Anwendungen kommunizieren miteinander über den IBM MQ-Kanal, der von den MCAs gesteuert wird. Für Anwendungen entspricht dieser Prozess dem Einreihen von Nachrichten in eine lokale Warteschlange.

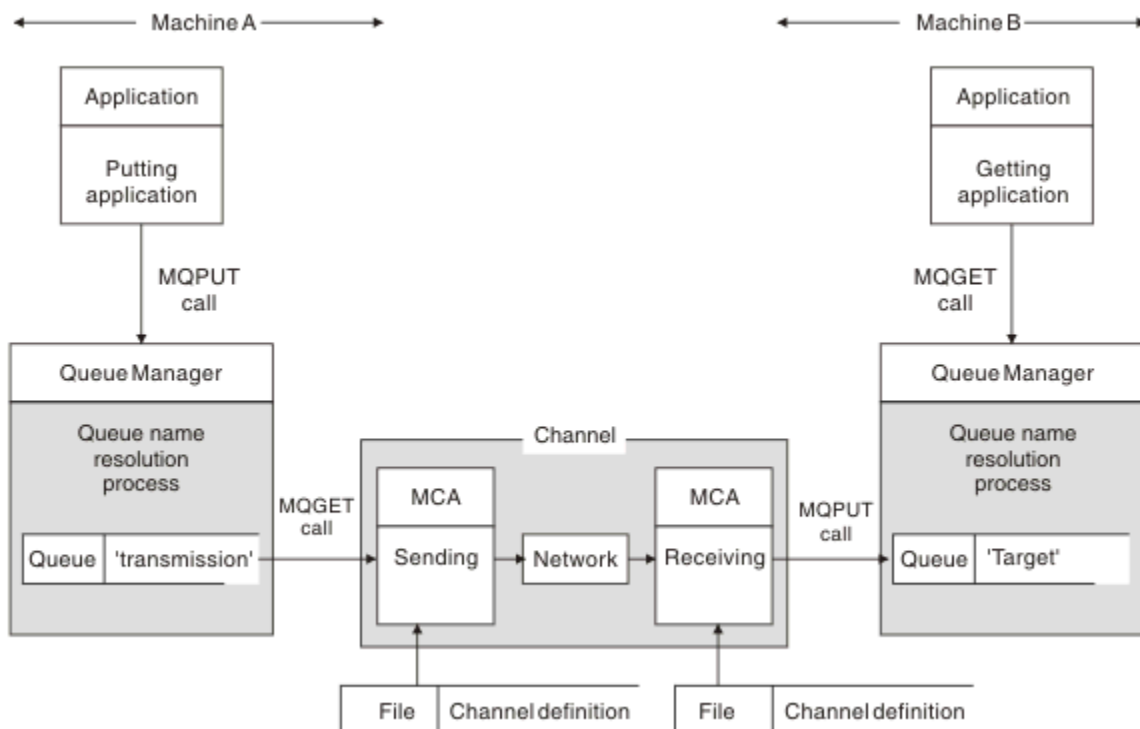


Abbildung 5. Namensauflösung

Wie in [Abbildung 5 auf Seite 81](#) beschrieben, ist der grundlegende Mechanismus für die Einreihung von Nachrichten in eine ferne Warteschlange in Bezug auf die Anwendung mit dem Prozess identisch, der für die Einreihung von Nachrichten in eine lokale Warteschlange genutzt wird:

- Die Anwendung, von der die Nachricht eingereicht wird, gibt zur Einreihung der Nachrichten in die Zielwarteschlange MQOPEN- und MQPUT-Aufrufe aus.
- Die Anwendung, von der die Nachricht abgerufen wird, gibt zum Abruf der Nachrichten aus der Zielwarteschlange MQOPEN- und MQGET-Aufrufe aus.

Sind beide Anwendungen mit demselben Warteschlangenmanager verbunden, ist keine Kommunikation zwischen Warteschlangenmanagern erforderlich, und die Zielwarteschlange wird beiden Anwendungen als *lokal* beschrieben.

Sind die Anwendungen jedoch mit unterschiedlichen Warteschlangenmanagern verbunden, werden zwei Nachrichtenkanalagenten mit ihrer zugehörigen Netzverbindung wie in der [Abbildung](#) dargestellt in die Übertragung einbezogen. In diesem Fall gilt die Zielwarteschlange bei der einreihenden Anwendung als *ferne Warteschlange*.

Der Ablauf der Ereignisse lautet wie folgt:

1. Die Anwendung, von der die Nachricht eingereicht wird, gibt zur Einreihung der Nachrichten in die Zielwarteschlange MQOPEN- und MQPUT-Aufrufe aus.
2. Während des MQOPEN-Aufrufs stellt die *Namensauflösung* fest, dass die Zielwarteschlange keine lokale Warteschlange ist, und entscheidet, welche Übertragungswarteschlange geeignet ist. Anschließend werden bei den MQPUT-Aufrufen, die dem MQOPEN-Aufruf zugeordnet sind, alle Nachrichten in diese Übertragungswarteschlange eingereiht.
3. Der sendende Nachrichtenkanalagent ruft die Nachrichten aus der Übertragungswarteschlange ab und übergibt sie beim fernen Computer an den empfangenden Nachrichtenkanalagenten.
4. Der empfangende Nachrichtenkanalagent reiht die Nachrichten in die Zielwarteschlange(n) ein.
5. Die Anwendung, von der die Nachricht abgerufen wird, gibt zum Abruf der Nachrichten aus der Zielwarteschlange MQOPEN- und MQGET-Aufrufe aus.

**Anmerkung:** Nur Schritt 1 und 5 enthalten Anwendungscode, während die Schritte 2 bis 4 von den lokalen Warteschlangenmanagern und Nachrichtenkanalagentenprogrammen ausgeführt werden. Die einreihende Anwendung kennt die Position der Zielwarteschlange nicht; diese kann sich in demselben Prozessor oder in einem anderen Prozessor auf einem anderen Kontinent befinden.

Die Kombination aus einem sendenden Nachrichtenkanalagenten, der Netzverbindung und dem empfangenden Nachrichtenkanalagenten wird als *Nachrichtenkanal* bezeichnet und ist grundsätzlich eine unidirektionale Einheit. Da Nachrichten für gewöhnlich in beide Richtungen fließen müssen, werden für diese Bewegung zwei Kanäle eingerichtet, und zwar einer für jeweils eine Richtung.

### Zugehörige Tasks

[Einreihen von Nachrichten in ferne Warteschlangen](#)

## Was ist die Auflösung von Warteschlangennamen?

Die Warteschlangennamensauflösung ist notwendig für verteiltes Warteschlangenmanagement. Anwendungen müssen die physische Adresse von Warteschlangen nicht mehr kennen und sie werden von den Netzwerkdetails isoliert.

Der Systemadministrator kann Warteschlangen von einem Warteschlangenmanager zu einem anderen verschieben und das Routing zwischen Warteschlangenmanagern ändern, ohne dass Anwendungen darüber informiert werden müssen.

Um den genauen Pfad, über den Daten weitergeleitet werden, vom Anwendungsdesign zu entkoppeln, gibt es eine Zwischenstufe zwischen dem Namen, der von der Anwendung für die Zielwarteschlange verwendet wird, und der Benennung des Kanals, über den der Datenfluss stattfindet. Diese Zwischenstufe wird durch die Funktion zur Auflösung des Warteschlangennamens erreicht.

Wenn eine Anwendung auf einen Warteschlangennamen verweist, wird der Name durch den Auflösungsmechanismus entweder einer Übertragungswarteschlange oder einer lokalen Warteschlange zugeordnet, die keine Übertragungswarteschlange ist. Für das Zuordnen zu einer Übertragungswarteschlange ist am Ziel eine zweite Namensauflösung erforderlich, dann wird die empfangene Nachricht in der Zielwarteschlange platziert, wie vom Anwendungsentwickler geplant. Die Anwendung hat keine Informationen zur Übertragungswarteschlange und zum Kanal, der zum Verschieben der Nachricht verwendet wurde.

**Anmerkung:** Für die Definition der Warteschlange und des Kanals ist das Systemmanagement zuständig. Sie können durch einen Operator oder ein Dienstprogramm für das Systemmanagement geändert werden, ohne dass die Anwendungen geändert werden müssen.

Eine wichtige Anforderung für das Systemmanagement von Nachrichtenflüssen besteht darin, zwischen Warteschlangenmanagern alternative Pfade bereitzustellen. So kann beispielsweise durch Geschäftsanforderungen festgelegt werden, dass unterschiedliche *Serviceklassen* über verschiedene Kanäle an dasselbe Ziel gesendet werden sollen. Dies ist eine Entscheidung des Systemmanagements, für die der Mechanismus zur Auflösung des Warteschlangennamens flexible Möglichkeiten bietet. Dies wird im Application Programming Guide detailliert erläutert. Die grundlegende Idee besteht jedoch darin, beim sendenden Warteschlangenmanager die Auflösung von Warteschlangennamen zu verwenden, um den von der Anwendung angegebenen Warteschlangennamen der für den jeweiligen Datenverkehrstyp geeigneten Übertragungswarteschlange zuzuordnen. Auf der Empfangsseite wird analog dazu durch die Auflösung von Warteschlangennamen der Name im Nachrichtendeskriptor einer lokalen Warteschlange (keiner Übertragungswarteschlange) zugeordnet oder erneut einer geeigneten Übertragungswarteschlange zugeordnet.

Nicht nur der Weiterleitungspfad von einem Warteschlangenmanager zu einem anderen kann in verschiedene Datenverkehrstypen partitioniert werden, sondern auch die Antwortnachricht, die in der abgehenden Nachricht an die Definition der Empfangswarteschlange für Antworten gesendet wird, kann dieselbe Datenverkehrspartitionierung verwenden. Die Auflösung des Warteschlangennamens erfüllt diese Anforderung und der Anwendungsentwickler muss in diese Entscheidungen zur Partitionierung des Datenverkehrs nicht einbezogen werden.

Die Tatsache, dass die Zuordnung vom sendenden und empfangenden Warteschlangenmanager ausgeführt wird, ist ein wichtiger Aspekt bei der Namensauflösung. Durch diese Zuordnung kann der Warteschlangename, der durch die Anwendung angegeben wird, die den PUT-Vorgang durchgeführt hat, beim sendenden Warteschlangenmanager einer lokalen Warteschlange oder einer Übertragungswarteschlange und anschließend beim empfangenden Warteschlangenmanager erneut einer lokalen oder einer Übertragungswarteschlange zugeordnet werden.

Für Antwortnachrichten von empfangenden Anwendungen oder MCAs wird die Auflösung auf dieselbe Weise ausgeführt, sodass das Rückgaberrouting über bestimmte Pfade mit Warteschlangendefinitionen bei allen Warteschlangenmanagern entlang der Route ermöglicht wird.

## Wie Zielobjektattribute für Aliasnamen, ferne Warteschlangen und Clusterwarteschlangen aufgelöst werden

Wenn die Namensauflösung für einen Anwendungs-API-Aufruf durchgeführt wird, werden Attribute, die sich auf die Verwendung des Objekts auswirken, aus einer Kombination aus ursprünglich benanntem Objekt, dem "Pfad" (siehe „Auflösung des Warteschlangennamens“ auf Seite 81), und aufgelöstem Zielobjekt aufgelöst. In einem Warteschlangenmanagercluster ist das fragliche "benannte Objekt" die Definition des Objekts im Cluster (Warteschlange oder Thema). Dies ist eine Untergruppe der Objektattribute, die von Warteschlangenmanagern gemeinsam genutzt werden und sichtbar sind, z. B. über **DISPLAY QCLUSTER**.

Wo ein Attribut für das benannte Objekt, das von der Anwendung geöffnet wurde, definiert werden kann, hat es Vorrang. Beispielsweise können alle DEF\*\*\*\*-Attribute (Standardpersistenz, Priorität und asynchrone PUT-Antwort) für Definitionen von Alias- und fernen Warteschlangen konfiguriert werden. Sie werden wirksam, wenn die Alias- oder ferne Warteschlange von einer Anwendung geöffnet wird, statt einer aufgelösten Zielwarteschlange oder Übertragungswarteschlange.

Attribute, die die Anwendungsinteraktion mit einem Zielobjekt einschränken oder begrenzen sollen, können in der Regel nicht für das benannte Objekt (Definition der fernen Warteschlange oder Aliasname)

definiert werden. Zum Beispiel können **MAXMSGL** und **MAXDEPTH** nicht für eine Definition einer fernen Warteschlange oder einen Aliasnamen festgelegt werden und sie werden nicht zwischen den Mitgliedern eines Warteschlangenmanagerclusters übergeben. Diese Attribute werden daher von der aufgelösten Warteschlange übernommen (z. B. der lokalen Warteschlange, der entsprechenden Übertragungswarteschlange oder **SYSTEM.CLUSTER.TRANSMIT.QUEUE**). Bei der Ankunft auf einem fernen Warteschlangenmanager kann für die Zustellung an die Zielwarteschlange eine zweite Einschränkung wirksam werden, die dazu führen kann, dass eine Nachricht in eine Warteschlange für nicht zustellbare Nachrichten gestellt wird oder der Kanal zwangsweise gestoppt wird.

Es ist zu beachten, dass die **PUT**- und **GET**-Aktivierung einen Sonderfall von Attributauflösung darstellen. Für beide Attribute gilt, dass eine Instanz von **DISABLED** im Warteschlangenpfad bewirkt, dass es ein allgemeines aufgelöstes Attribut **DISABLED** gibt.

## System- und Standardobjekte

Hier sind die über den Befehl **crtmqm** erstellten System- und Standardobjekte aufgeführt.

Wenn Sie mit dem Steuerbefehl **crtmqm** einen Warteschlangenmanager erstellen, werden die System- und Standardobjekte automatisch erstellt.

- Systemobjekte sind die für den Betrieb eines Warteschlangenmanagers oder Kanals erforderlichen IBM MQ-Objekte.
- In den Standardobjekten sind alle Attribute eines Objekts definiert. Wenn Sie ein Objekt erstellen (z. B. eine lokale Warteschlange), werden alle Attribute, die Sie nicht explizit angeben, aus dem Standardobjekt übernommen.

In den folgenden Tabellen sind die System- und Standardobjekte aufgeführt, die von **crtmqm** erstellt werden.

**Anmerkung:** Es gibt zwei weitere Standardobjekte, die nicht in den Tabellen enthalten sind: das Warteschlangenmanagerobjekt und der Objektkatalog. Diese Objekte werden protokolliert und sind wiederherstellbar.

- [System- und Standardobjekte: Warteschlangen](#)
- [System- und Standardobjekte: Themen](#)
- [System- und Standardobjekte: Serverkanäle](#)
- [System- und Standardobjekte: Clientkanäle](#)
- [System- und Standardobjekte: Authentifizierungsdaten](#)
- [System- und Standardobjekte: Kommunikationsdaten](#)
- [System- und Standardobjekte: Empfangsprogramme](#)
- [System- und Standardobjekte: Namenslisten](#)
- [System- und Standardobjekte: Prozesse](#)
- [System- und Standardobjekte: Services](#)

<i>Tabelle 12. System- und Standardobjekte: Warteschlangen</i>	
<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.ADMIN.ACCOUNTING.QUEUE	Warteschlange für Abrechnungsnachrichtendaten, die bei der Trennung der Verbindung zwischen einer Anwendung und dem Warteschlangenmanager generiert werden.
SYSTEM.ADMIN.ACTIVITY.QUEUE	Die Warteschlange für zurückgegebene Aktivitätsberichts-nachrichten.
SYSTEM.ADMIN.CHANNEL.EVENT	Ereigniswarteschlange für Kanäle
SYSTEM.ADMIN.COMMAND.EVENT	Ereigniswarteschlange für Befehlsereignisse.

Tabelle 12. System- und Standardobjekte: Warteschlangen (Forts.)

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.ADMIN.COMMAND.QUEUE	Warteschlange für Verwaltungsbefehle. Wird für ferne MQSC-Befehle und PCF-Befehle verwendet.
SYSTEM.ADMIN.CONFIG.EVENT	Ereigniswarteschlange für Konfigurationsereignisse.
SYSTEM.ADMIN.LOGGER.EVENT	Ereigniswarteschlange für Protokollierungsereignisnachrichten (Journalempfänger).
SYSTEM.ADMIN.PERFM.EVENT	Ereigniswarteschlange für Leistungsereignisse
SYSTEM.ADMIN.PUBSUB.EVENT	Ereigniswarteschlange für Publish/Subscribe-Ereignisse des Systems
SYSTEM.ADMIN.QMGR.EVENT	Ereigniswarteschlange für Warteschlangenmanagereignisse.
SYSTEM.ADMIN.STATISTICS.QUEUE	Die Warteschlange für MQI-, Warteschlangen- und Kanalstatistiknachrichtendaten.
SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE	Die Warteschlange für die Anzeige der Trace-Aktivitäten.
SYSTEM.ADMIN.TRACE.ROUTE.QUEUE	Die Warteschlange für zurückgegebene Trace-Route-Antwortnachrichten.
SYSTEM.AUTH.DATA.QUEUE	Warteschlange mit den Zugriffskontrolllisten für den Warteschlangenmanager Wird vom Objektberechtigungsmanager verwendet
SYSTEM.BROKER.CONTROL.QUEUE	Steuerwarteschlange für Publish/Subscribe-Schnittstelle
SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS	Warteschlange für Kommunikation zwischen Brokern
SYSTEM.CHANNEL.INITQ	Kanalinitialisierungswarteschlange
SYSTEM.CHANNEL.SYNCQ	Warteschlange für die Synchronisationsdaten für Kanäle
SYSTEM.CHLAUTH.DATA.QUEUE	Warteschlange mit IBM MQ-Kanalauthentifizierungsdaten
SYSTEM.DURABLE.MODEL.QUEUE	Die Warteschlange, die als Modell für verwaltete, permanente Subskriptionen verwendet wird
SYSTEM.DURABLE.SUBSCRIBER.QUEUE	Warteschlange mit einer persistenten Kopie der permanenten Subskriptionen des Warteschlangenmanagers
SYSTEM.CICS.INITIATION.QUEUE	CICS-Standardinitialisierungswarteschlange.
SYSTEM.CLUSTER.COMMAND.QUEUE	Warteschlange für die Übertragung von Nachrichten an den Repository-Warteschlangenmanager
SYSTEM.CLUSTER.HISTORY.QUEUE	Die Warteschlange für die Speicherung des Protokolls von Clusterstatusinformationen zu Servicezwecken
SYSTEM.CLUSTER.REPOSITORY.QUEUE	Warteschlange für die Speicherung aller Repositoryinformationen

Tabelle 12. System- und Standardobjekte: Warteschlangen (Forts.)

Objektname	Beschreibung
SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE	Die Warteschlange zum Erstellen der einzelnen Übertragungswarteschlangen für den jeweiligen Cluster-senderkanal
SYSTEM.CLUSTER.TRANSMIT.QUEUE	Übertragungswarteschlange für alle Nachrichten an alle Cluster
SYSTEM.DEAD.LETTER.QUEUE	Warteschlange für nicht zustellbare Nachrichten.
SYSTEM.DEFAULT.ALIAS.QUEUE	Standardaliaswarteschlange
SYSTEM.DEFAULT.INITIATION.QUEUE	Standardinitialisierungswarteschlange
SYSTEM.DEFAULT.LOCAL.QUEUE	Standardmäßige lokale Warteschlange
SYSTEM.DEFAULT.MODEL.QUEUE	Standardmodellwarteschlange
SYSTEM.DEFAULT.REMOTE.QUEUE	Standardmäßige ferne Warteschlange
SYSTEM.HIERARCHY.STATE	Hierarchiebeziehungsstatus für verteiltes Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.CONTROL	Steuerwarteschlange für verteiltes Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.FANREQ	Eingabewarteschlange für Ausgabefächerungsprozesse der internen Proxy-Subskription des verteilten Publish/Subscribe von IBM MQ.
SYSTEM.INTER.QMGR.PUBS	Veröffentlichungen des verteilten Publish/Subscribe von IBM MQ.
SYSTEM.JMS.TEMPQ.MODEL	Modell für temporäre JMS-Warteschlangen
SYSTEM.MQEXPLORER.REPLY.MODEL	Die IBM MQ Explorer-Warteschlange für zu beantwortende Nachrichten. Hierbei handelt es sich um eine Modellwarteschlange, mit der eine temporäre dynamische Warteschlange für Antworten an den IBM MQ Explorer erstellt wird.
SYSTEM.MQSC.REPLY.QUEUE	Empfangswarteschlange für Antworten für MQSC-Befehle. Dies ist eine Modellwarteschlange, aus der eine temporäre dynamische Warteschlange für Antworten an ferne MQSC-Befehle erstellt wird.
SYSTEM.NDURABLE.MODEL.QUEUE	Warteschlange, die als Modell für verwaltete, nicht permanente Subskriptionen verwendet wird
SYSTEM.PENDING.DATA.QUEUE	Unterstützt verzögerte Nachrichten in JMS.
SYSTEM.RETAINED.PUB.QUEUE	Warteschlange für eine Kopie jeder ständigen Veröffentlichung im Warteschlangenmanager

Tabelle 13. System- und Standardobjekte: Themen

Objektname	Beschreibung
SYSTEM.ADMIN.TOPIC	Verwaltungsthema.

*Tabelle 13. System- und Standardobjekte: Themen (Forts.)*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.BASE.TOPIC	Basisthema für die ASPARENT-Auflösung. Gibt es zu einem bestimmten Thema keine übergeordneten Verwaltungsthemenobjekte oder weisen diese übergeordneten Objekte auch den Wert ASPARENT auf, werden alle verbleibenden ASPARENT-Attribute von diesem Objekt übernommen.
SYSTEM.BROKER.ADMIN.STREAM	Verwaltungsdatenstrom, der von der Schnittstelle für eingereihtes Publish/Subscribe verwendet wird
SYSTEM.BROKER.DEFAULT.STREAM	Standarddatenstrom, der von der Schnittstelle für eingereihtes Publish/Subscribe verwendet wird
SYSTEM.BROKER.DEFAULT.SUBPOINT	Der Standardunterpunkt, der von der Schnittstelle für eingereihtes Publish/Subscribe verwendet wird
SYSTEM.DEFAULT.TOPIC	Standardthemendefinition.

*Tabelle 14. System- und Standardobjekte: Serverkanäle*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.AUTO.RECEIVER	Dynamischer Empfängerkanal
SYSTEM.AUTO.SVRCONN	Dynamischer Serververbindungskanal
SYSTEM.DEF.CLUSRCVR	Standardmäßiger Empfängerkanal für den Cluster zur Bereitstellung von Standardwerten für alle Attribute, die bei der Erstellung eines CLUSRCVR-Kanals auf einem Warteschlangenmanager im Cluster nicht angegeben werden.
SYSTEM.DEF.CLUSSDR	Standardmäßiger Senderkanal für den Cluster zur Bereitstellung von Standardwerten für alle Attribute, die bei der Erstellung eines CLUSSDR-Kanals auf einem Warteschlangenmanager im Cluster nicht angegeben werden.
SYSTEM.DEF.RECEIVER	Standardmäßiger Empfängerkanal
SYSTEM.DEF.REQUESTER	Standardmäßiger Requesterkanal
SYSTEM.DEF.SENDER	Standardmäßiger Senderkanal
SYSTEM.DEF.SERVER	Standardmäßiger Serverkanal
SYSTEM.DEF.SVRCONN	Standardmäßiger Serververbindungskanal
SYSTEM.DEF.AMQP	AMQP-Standardkanal. Beachten Sie, dass das Objekt definiert ist, aber der AMQP-Service nicht unterstützt wird.

*Tabelle 15. System- und Standardobjekte: Clientkanäle*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEF.CLNTCONN	Standardclientverbindungskanal.



*Tabelle 16. System- und Standardobjekte: Authentifizierungsdaten*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.AUTHINFO.CRLLDAP	Standardauthentifizierungsdatenobjekt zur Definition von Authentifizierungsdatenobjekten vom Typ CRLLDAP.
SYSTEM.DEFAULT.AUTHINFO.OCSP	Standardauthentifizierungsdatenobjekt zur Definition von Authentifizierungsdatenobjekten vom Typ OCSP.

*Tabelle 17. System- und Standardobjekte: Kommunikationsdaten*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.COMMINFO.MULTICAST	Standardkommunikationsinformationsobjekt für Multicasting.

*Tabelle 18. System- und Standardobjekte: Empfangsprogramme*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.LISTENER.TCP	Standardempfangsprogramm für TCP-Transport
 SYSTEM.DEFAULT.LISTENER.LU62	LU62-Standardempfangsprogramm.
 SYSTEM.DEFAULT.LISTENER.NETBIOS	NETBIOS-Standardempfangsprogramm.
 SYSTEM.DEFAULT.LISTENER.SPX	SPX-Standardempfangsprogramm.

*Tabelle 19. System- und Standardobjekte: Namenslisten*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.NAMELIST	Standardnamenslistendefinition
SYSTEM.QPUBSUB.QUEUE.NAMELIST	Liste mit Warteschlangennamen, die von der Schnittstelle für eingereichtes Publish/Subscribe überwacht werden
SYSTEM.QPUBSUB.SUBPOINT.NAMELIST	Liste mit Themenobjekten, die von der Schnittstelle für eingereichtes Publish/Subscribe zum Abgleichen von Themenobjekten mit Subskriptionspunkten verwendet werden

*Tabelle 20. System- und Standardobjekte: Prozesse*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.DEFAULT.PROCESS	Standardprozessdefinition.

*Tabelle 21. System- und Standardobjekte: Services*

<b>Objektname</b>	<b>Beschreibung</b>
SYSTEM.AMQP.SERVICE	MQ Light API-Service. Beachten Sie, dass das Objekt definiert ist, aber der Service nicht unterstützt wird.



Tabelle 21. System- und Standardobjekte: Services (Forts.)

Objektname	Beschreibung
SYSTEM.DEFAULT.SERVICE	Standardservice.

## Windows Objekte der Windows-Standardkonfiguration

Auf Windows-Systemen können Sie mithilfe der Anwendung IBM MQ Postcard eine Standardkonfiguration einrichten.

**Anmerkung:** Wenn auf Ihrem Computer andere Warteschlangenmanager vorhanden sind, können Sie keine Standardkonfiguration einrichten.

Für viele Namen, die für Objekte der Windows-Standardkonfiguration verwendet werden, wird ein TCP/IP-Kurzname verwendet. Dies ist der TCP/IP-Name des Computers ohne die Domänenkomponente. Der TCP/IP-Kurzname des Computers `mycomputer.hursley.ibm.com` beispielsweise lautet `mycomputer`. Immer wenn dieser Name abgeschnitten werden muss und das letzte Zeichen ein Punkt (.) ist, wird dieser entfernt.

Alle Zeichen in dem TCP/IP-Kurznamen, die für IBM MQ-Objektnamen nicht gültig sind (beispielsweise Bindestriche) werden durch ein Unterstrichungszeichen ersetzt.

Gültige Zeichen für IBM MQ -Objektnamen sind: a bis z, A bis Z, 0 bis 9 und die vier Sonderzeichen/%, und \_.

Der Clustername für die Windows-Standardkonfiguration ist `DEFAULT_CLUSTER`.

Wenn es sich bei dem Warteschlangenmanager nicht um einen Repository-Warteschlangenmanager handelt, werden die in [Tabelle 22 auf Seite 89](#) aufgeführten Objekte erstellt.

Tabelle 22. Von der Anwendung für die Windows-Standardkonfiguration erstellte Objekte

Objekt	Name
Warteschlangenmanager	Der TCP/IP-Kurzname, dem die Zeichen <code>QM_</code> vorangestellt sind. Die maximale Länge des Namens des Warteschlangenmanagers beträgt 48 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 48 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.  Dem Warteschlangenmanager sind ein Befehlsserver, ein Kanalempfangsprogramm und ein Kanalinitiator zugeordnet. Das Kanalempfangsprogramm überwacht den IBM MQ-Standardport mit der Portnummer 1414. Alle anderen auf diesem System erstellten Warteschlangenmanager dürfen Port 1414 nicht verwenden, solange der Warteschlangenmanager der Standardkonfiguration vorhanden ist.
Generischer Clusterempfängerkanal	Der TCP/IP-Kurzname mit dem Präfix <code>TO_QM_</code> . Die maximale Länge des generischen Clusterempfängernamens beträgt 20 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 20 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.
Clustersenderkanal	Der Clustersenderkanal wird anfänglich mit dem Namen <code>TO_ + QMNAME +</code> erstellt. Wenn IBM MQ eine Verbindung zum Repository-Warteschlangenmanager für den Standardkonfigurationscluster hergestellt hat, wird dieser Name durch den Namen des Repository-Warteschlangenmanagers für den Standardkonfigurationscluster mit dem Präfix <code>TO_</code> ersetzt. Die maximale Länge des Clustersenderkanalnamens beträgt 20 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 20 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.

Tabelle 22. Von der Anwendung für die Windows-Standardkonfiguration erstellte Objekte (Forts.)

Objekt	Name
Lokale Nachrichtenwarteschlange	Die lokale Nachrichtenwarteschlange hat die Bezeichnung default.
Lokale Nachrichtenwarteschlange zur Verwendung durch die Anwendung IBM MQ Postcard	Die lokale Nachrichtenwarteschlange zur Verwendung durch die Anwendung IBM MQ Postcard hat die Bezeichnung postcard.
Serververbindungskanal	Mithilfe des Serververbindungskanals können Clients eine Verbindung zum Warteschlangenmanager herstellen. Der Name ist der TCP/IP-Kurzname, dem die Zeichen S_ vorangestellt sind. Die maximale Länge des Serververbindungskanalnamens beträgt 20 Zeichen. Namen, die diesen Grenzwert überschreiten, werden nach 20 Zeichen abgeschnitten. Wenn das letzte Zeichen des Namens ein Punkt (.) ist, wird dieser durch ein Leerzeichen ( ) ersetzt.

Wenn es sich bei dem Warteschlangenmanager um einen Repository-Warteschlangenmanager handelt, ähnelt die Standardkonfiguration der in [Tabelle 22 auf Seite 89](#) beschriebenen Konfiguration, weist jedoch folgende Unterschiede auf:

- Der Warteschlangenmanager wird als Repository-Warteschlangenmanager für den Cluster der Standardkonfiguration definiert.
- Es wird kein Clustersenderkanal definiert.
- Es wird eine lokale Clusterwarteschlange erstellt, deren Name der TCP/IP-Kurzname mit den vorangestellten Zeichen clq\_default\_ ist. Die maximale Länge dieses Namens liegt bei 48 Zeichen. Namen, die diese Länge überschreiten, werden nach 48 Zeichen abgeschnitten.

Wenn Sie Einrichtungen zur Fernverwaltung anfordern, wird außerdem der Serververbindungskanal SYSTEM.ADMIN.SVRCONN erstellt.



## SYSTEM.BASE.TOPIC

Basisthema für die ASPARENT-Auflösung. Gibt es zu einem bestimmten Thema keine übergeordneten Verwaltungsthemenobjekte oder weisen diese übergeordneten Objekte auch den Wert ASPARENT auf, werden alle verbleibenden ASPARENT-Attribute von diesem Objekt übernommen.

Tabelle 23. Standardwerte von SYSTEM.BASE.TOPIC

Parameter	Wert
TOPICSTR	"
CLROUTE	DIRECT
Cluster	Der Standardwert ist eine leere Zeichenfolge.
COMMINFO	SYSTEM.DEFAULT.COMMINFO.MULTICAST
DEFPRESP	SYNC
DEFPRTY	0
DEFPSIST	NO
DESCR	'Base topic for resolving attributes'
DURSUB	YES
MCAST	DISABLED
MDURMDL	SYSTEM.DURABLE.MODEL.QUEUE

Tabelle 23. Standardwerte von SYSTEM.BASE.TOPIC (Forts.)

Parameter	Wert
MNDURMDL	SYSTEM.NDURABLE.MODEL.QUEUE
NPMSGDLV	ALLAVAIL
PMSGDLV	ALLDUR
PROXYSUB	FIRSTUSE
PUB	ENABLED
PUBSCOPE	ALL
  QSGDISP (nur z/OS-Plattform)	QMGR
SUB	ENABLED
SUBSCOPE	ALL
USEDLQ	YES
WILDCARD	PASSTHRU

Falls dieses Objekt nicht vorhanden ist, werden seine Standardwerte von IBM MQ dennoch für ASPARENT-Attribute verwendet, die nicht über übergeordnete Themen in der Themenstruktur aufgelöst werden.

Durch Einstellen des Attributs PUB oder SUB von SYSTEM.BASE.TOPIC auf DISABLED wird verhindert, dass Anwendungen Themen in der Themenstruktur publizieren oder subscribieren, mit zwei Ausnahmen:

1. Themenobjekte in der Themenstruktur, deren Attribut PUB oder SUB explizit auf ENABLE eingestellt ist. Anwendungen können diese Themen und die ihnen untergeordneten Elemente publizieren und subscribieren.
2. Die Publizierung und Subskription von SYSTEM.BROKER.ADMIN.STREAM wird durch die Einstellung des Attributs PUB oder SUB von SYSTEM.BASE.TOPIC auf DISABLED nicht inaktiviert.


Weitere Informationen finden Sie unter [Besondere Handhabung des Parameters PUB](#).



## Informationen zu Zeilengruppen


Anhand der folgenden Informationen können Sie die Angaben innerhalb von Zeilengruppen konfigurieren. Zudem finden Sie hier eine Auflistung des Inhalts der Dateien `mqc.ini`, `qm.ini` und `mqclient.ini`.

### Zeilengruppen konfigurieren

Rufen Sie über die folgenden Links hilfreiche Informationen zur Konfiguration des Systems bzw. der Systeme in Ihrem Unternehmen auf:

- [IBM MQ-Konfigurationsdaten ändern](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *AllQueueManagers*
  - Zeilengruppe *DefaultQueueManager*
  - Zeilengruppe *ExitProperties*
  - Zeilengruppe *LogDefaults*
  - Zeilengruppe *Security* in der Datei `qm.ini`
- [Konfigurationsdaten von Warteschlangenmanagern ändern](#) enthält Informationen zur Konfiguration der folgenden Elemente:
  -  Zeilengruppe *AccessMode* (nur unter Windows)

- Zeilengruppe *Service* - für installierbare Services
- Zeilengruppe *Log*
-   Zeilengruppe *RestrictedMode* (nur auf Systemen mit UNIX and Linux)
- Zeilengruppe *XAResourceManager*
- Zeilengruppen *TCP*, *LU62* und *NETBIOS*
- Zeilengruppe *ExitPath*
- Zeilengruppe *QMErrorLog*
- Zeilengruppe *SSL*
- Zeilengruppe *ExitPropertiesLocal*
- Services und Komponenten konfigurieren enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *Service*
  - Zeilengruppe *ServiceComponent*

Außerdem finden Sie dort Links zu Informationen über deren Verwendung für die verschiedenen Services unter UNIX and Linux und auf Windows-Plattformen.
- API-Exits konfigurieren enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *AllActivityTrace*
  - Zeilengruppe *ApplicationTrace*
- Verhalten des Aktivitätstrace konfigurieren enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *ApiExitCommon*
  - Zeilengruppe *ApiExitTemplate*
  - Zeilengruppe *APIExitLocal*
- Konfigurationsdaten für Clients enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *CHANNELS*
  - Zeilengruppe *ClientExitPath*
  -  Zeilengruppen *LU62*, *NETBIOS* und *SPX* (nur unter Windows)
  - Zeilengruppe *MessageBuffer*
  - Zeilengruppe *SSL*
  - Zeilengruppe *TCP*
- „Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen“ auf Seite 94 enthält Informationen zur Konfiguration der folgenden Elemente:
  - Zeilengruppe *CHANNELS*
  - Zeilengruppe *TCP*
  - Zeilengruppe *LU62*
  - *NETBIOS*
  - Zeilengruppe *ExitPath*
- Attribute von eingereichten Publish/Subscribe-Nachrichten festlegen enthält Informationen zur Konfiguration der folgenden Elemente:
  - Attribut *PersistentPublishRetry*
  - Attribut *NonPersistentPublishRetry*
  - Attribut *PublishBatchSize*
  - Attribut *PublishRetryInterval*

in der Zeilengruppe *Broker*.





**Achtung:** Falls Sie eine *Broker*-Zeilengruppe benötigen, müssen Sie diese erstellen.

- **V 9.1.4** Mithilfe der automatischen Konfiguration können Sie Folgendes konfigurieren:
  - Zeilengruppe 'AutoConfig'
  - Zeilengruppe 'AutoCluster'
  - Zeilengruppe 'Variables'

## Konfigurationsdateien

Unter

- Datei [mq<sub>s</sub>.ini](#)
- Datei [qm.ini](#)
- Datei [mqclient.ini](#)

Dort finden Sie eine Liste der möglichen Zeilengruppen in den einzelnen Konfigurationsdateien.  

### mq<sub>s</sub>.ini-Datei

Ein Beispiel für eine Datei `mqs.ini` finden Sie im Abschnitt [Beispiel einer IBM MQ-Konfigurationsdatei für UNIX and Linux-Systeme](#).

Eine Datei `mqs.ini` kann die folgenden Zeilengruppen enthalten:

- [AllQueueManagers](#)
- [DefaultQueueManager](#)
- [ExitProperties](#)
- [LogDefaults](#)

Außerdem gibt es für jeden Warteschlangenmanager eine [QueueManager](#)-Zeilengruppe.

### Datei 'qm.ini'

Unter [Beispiel einer Konfigurationsdatei für einen Warteschlangenmanager in IBM MQ für Systeme mit UNIX and Linux](#) finden Sie ein Beispiel für die Datei `qm.ini`.

Eine Datei `qm.ini` kann die folgenden Zeilengruppen enthalten:

- [ExitPath](#)
- [Protokoll](#)
- [QMErrorLog](#)
- [QueueManager](#)
- [Sicherheit](#)
- [ServiceComponent](#)



Verwenden Sie zum Konfigurieren von [InstallableServices](#) die Zeilengruppen `Service` und `ServiceComponent`.

- `Connection` für [DefaultBindType](#)






**Achtung:** Falls Sie eine `Connection`-Zeilengruppe benötigen, müssen Sie diese erstellen.

- [SSL und TLS](#)
- [TCP, LU62 und NETBIOS](#)

- [XAResourceManager](#)

Zusätzlich können Sie Folgendes ändern:

-  [AccessMode](#) (nur unter Windows)
-   [RestrictedMode](#) (nur auf Systemen mit UNIX and Linux)

Hierfür verwenden Sie den Befehl [crtmqm](#).

### Datei 'mqclient.ini'

Eine Datei `mqclient.ini` kann die folgenden Zeilengruppen enthalten:

- [CHANNELS](#)
- [ClientExitPath](#)
- [LU62, NETBIOS und SPX](#)
- [MessageBuffer](#)
- [SSL](#)
- [TCP](#)

Darüber hinaus wird möglicherweise eine [Zeilengruppe PreConnect](#) zur Konfiguration eines PreConnect-Exits benötigt.

## Zeilengruppen der Konfigurationsdatei für die verteilte Steuerung von Warteschlangen

Beschreibung der Zeilengruppen der Warteschlangenmanager-Konfigurationsdatei `qm.ini` für die verteilte Steuerung von Warteschlangen

Dieser Abschnitt enthält die Zeilengruppen in der Warteschlangenmanager-Konfigurationsdatei für die verteilte Steuerung von Warteschlangen. Sie gelten für die Warteschlangenmanager-Konfigurationsdatei für IBM MQ for Multiplatforms. Die Datei heißt auf allen Plattformen `qm.ini`.

Zeilengruppen für die verteilte Steuerung von Warteschlangen sind:

- Kanäle
- TCP
- LU62
- NETBIOS
- EXITPATH

In [Abbildung 6 auf Seite 95](#) sind die Werte aufgeführt, die Sie mit diesen Zeilengruppen festlegen können. Wenn Sie eine dieser Zeilengruppen definieren, müssen Sie nicht jedes Element in einer neuen Zeile beginnen. Sie können entweder ein Semikolon (;) oder ein Nummernzeichen (#) verwenden, um einen Kommentar anzugeben.

```

CHANNELS:
MAXCHANNELS=n           ; Maximum number of channels allowed, the
                        ; default value is 100.
MAXACTIVECHANNELS=n    ; Maximum number of channels allowed to be active at
                        ; any time, the default is the value of MaxChannels.
MAXINITIATORS=n        ; Maximum number of initiators allowed, the default
                        ; and maximum value is 3.
MQIBINDTYPE=type       ; Whether the binding for applications is to be
                        ; "fastpath" or "standard".
                        ; The default is "standard".
PIPELINELENGTH=n       ; The maximum number of concurrent threads a channel will use.
                        ; The default is 1. Any value greater than 1 is treated as 2.
ADOPTNEWMCA=chltype    ; Stops previous process if channel fails to start.
                        ; The default is "NO".
ADOPTNEWMCATIMEOUT=n   ; Specifies the amount of time that the new
                        ; process should wait for the old process to end.
                        ; The default is 60.
ADOPTNEWMCACHECHECK=   ; Specifies the type checking required.
  typecheck             ; The default is "NAME", "ADDRESS", and "QM".
CHLAUTHEARLYADOPT=Y/N ; The order in which connection authentication and channel authentica
tion rules are          ; processed. If not present in the qm.ini file the default is "N".
From MQ9.0.4 all
PASSWORDPROTECTION=    ; queue managers are created with a default of "Y"
  than using TLS.      ; From MQ8.0, set protected passwords in the MQCSP structure, rather
  options              ; The options are "compatible", "always", "optional" and "warn"
                        ; The default is "compatible".
IGNORESEQNUMBERMISMATCH ; How the queue manager handles a sequence number mismatch during chan
nel startup.          ;
  =Y/N                ; The options are "Y" and "N" with the default being "N".
CHLAUTHIGNOREUSERCASE  ; Enables a queue manager to make username matching within CHLAUTH
rules case-insensitive. ;
  =Y/N                ; The options are "Y" and "N" with the default being "N".
CHLAUTHISSUEWARN=Y     ; If you want message AMQ9787 to be generated when you set theWARN=YES
attribute              ;
TCP:                   ; on the SET CHLAUTH command.
  PORT=n               ; TCP entries
  KEEPALIVE=Yes        ; Port number, the default is 1414
  LU62:                ; Switch TCP/IP KeepAlive on
  LIBRARY2=DLLName2    ; Used if code is in two libraries
EXITPATH:1 Location of user exits
EXITPATHS=             ; String of directory paths.

```

Abbildung 6. Zeilengruppen der Datei `qm.ini` für die verteilte Steuerung von Warteschlangen


### Anmerkungen:

1. EXITPATH gilt nur auf den folgenden Plattformen:

-  AIX
-  Solaris
-  Windows

### Zugehörige Tasks

konfigurieren

 z/OS konfigurieren

Konfigurationsdaten auf Windows-, UNIX- und Linux-Systemen ändern.

 Konfigurationsdaten unter IBM i ändern

## Kanalattribute

In diesem Abschnitt werden die in den Kanaldefinitionen angegebenen Kanalattribute beschrieben.

Sie wählen die Attribute eines Kanals so aus, dass sie unter bestimmten Bedingungen für jeden Kanal optimal sind. Wenn der Kanal dann ausgeführt wird, wurden die tatsächlichen Werte jedoch möglicherweise

im Rahmen von Startvereinbarungen geändert. Weitere Informationen finden Sie im Abschnitt [Kanäle vorbereiten](#).

Für viele Attribute gibt es Standardwerte, die für die meisten Kanäle verwendet werden können. Für Bedingungen, unter denen die Standardwerte nicht optimal sind, finden Sie in diesem Abschnitt eine Anleitung zur Auswahl der richtigen Werte.

Für Clusterkanäle geben Sie die Clusterkanalattribute auf den Ziel-Warteschlangenmanagern auf den Clusterempfängerkanälen an. Attribute, die Sie auf den zugehörigen Clustersenderkanälen angeben, werden vermutlich ignoriert. Siehe [Clusterkanäle](#).

**Anmerkung:** In IBM MQ for IBM i können die meisten Attribute als \*SYSDFTCHL angegeben werden, d. h., es wird der Wert des Standardkanals Ihres Systems übernommen.

## Kanalattribute und Kanaltypen

Unterschiedliche Kanaltypen unterstützen unterschiedliche Kanalattribute.

Die Kanaltypen für IBM MQ-Kanalattribute werden in der folgenden Tabelle aufgelistet.

**Anmerkung:** Wenn bei Clusterkanälen (die Tabellenspalten CLUSSDR und CLUSRCVR) ein Attribut für beide Kanäle gesetzt werden kann, definieren Sie das Attribut für beide Kanäle und achten Sie darauf, dass für beide Kanäle dieselbe Attributeinstellung verwendet wird. Bei unterschiedlichen Einstellungen werden in der Regel die Einstellungen verwendet, die Sie für den Kanal CLUSRCVR angegeben haben. Dies wird im Abschnitt [Clusterkanäle](#) erläutert.


Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AMQP
<a href="#">Datum ändern</a>	ALTDATE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<a href="#">Zeit ändern</a>	ALTTIME	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
 <a href="#">AMQP-Keepalive</a>	AMQPKA									Ja
<a href="#">Intervall der Überwachungssignale für Stapel</a>	BATCHHB	Ja	Ja					Ja	Ja	
<a href="#">Stapelintervall</a>	BATCHINT	Ja	Ja					Ja	Ja	
<a href="#">Batch limit (Stapeldatengrenzwert)</a>	BATCHLIM	Ja	Ja					Ja	Ja	
<a href="#">Stapelgröße</a>	BATCHSZ	Ja	Ja	Ja	Ja			Ja	Ja	
<a href="#">Zertifikatsbezeichnung</a>	CERTLABL	Ja	Ja	Ja	Ja	Ja	Ja	Yes (Ja) <small>„1“ auf Seite 100</small>	Ja	Ja
<a href="#">Channel Name</a>	CHANNEL	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<a href="#">Kanalstatistik</a>	STATCHL	Ja	Ja	Ja	Ja			Ja	Ja	
<a href="#">Channel Type</a>	CHLTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<a href="#">Clientkanalge- wichtung</a>	CLNTWGH T					Ja				



Tabelle 24. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Be- fehlspara- meter	SDR	SVR	RCV R	RQST R	CLNT- CONN	SVR- CONN	CLUS - SDR	CLUS- RCVR	AM QP
<u>Cluster</u>	CLUSTER							Ja	Ja	
<u>Clusternamenslis- te</u>	CLUSNL							Ja	Ja	
<u>Clusterauslas- tungspriorität</u>	CLWLPRT Y							Ja	Ja	
<u>Clusterauslas- tungsrangordnung</u>	CLWLRAN K							Ja	Ja	
<u>Clusterauslas- tungsgewichtung</u>	CLWLWGH T							Ja	Ja	
<u>Verbindungsaffini- tät</u>	AFFINITY					Ja				
<u>Verbindungsname</u>	CONNNAME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Nachricht umwan- deln</u>	CONVERT	Ja	Ja					Ja	Ja	
<u>Datenkomprimie- rung</u>	COMPMSG	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Standardmäßige Verbindungswie- derholung</u>	DEFRE- CON					Ja				
<u>Beschreibung</u>	DESCR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>Unterbrechungs- intervall</u>	DISCINT	Ja	Ja				Yes (Ja) „2“ auf Seite 100	Ja	Ja	
<u>Disposition „2“ auf Seite 100</u>	QSGDISP	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Header-Kompri- mierung</u>	COMPHDR	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Intervall der Über- wachungssignale</u>	HBINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Keepalive-Inter- vall</u>	KAINT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Lokale Adresse</u>	LOC- LADDR	Ja	Ja		Ja	Ja		Ja	Ja	Ja
<u>Zähler für Wieder- holungsversuche nach langem In- tervall</u>	LONGRTY	Ja	Ja					Ja	Ja	
<u>Langes Wiederho- lungsintervall</u>	LONGTMR	Ja	Ja					Ja	Ja	

Tabelle 24. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS - SDR	CLUS-RCVR	AM QP
<u>Gibt den LU 6.2-Modusnamen an.</u>	MODENAME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Gibt das LU 6.2-Transaktionsprogramm an.</u>	TPNAME	Ja	Ja		Ja	Ja		Ja	Ja	
<u>Maximale Instanzen</u>	MAXINST						Ja			Ja
<u>Maximale Instanzen pro Client</u>	MAXINSTC						Ja			
<u>Maximale Nachrichtenlänge</u>	MAXMSGL	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>Name des Nachrichtenkanalagenten</u>	MCANAME	Ja	Ja		Ja			Ja	Ja	
<u>Typ des Nachrichtenkanalagenten</u>	MCATYPE	Ja	Ja		Ja			Ja	Ja	
<u>Benutzer des Nachrichtenkanalagenten</u>	MCAUSER	Ja	Ja	Ja	Ja		Ja	Ja	Ja	Ja
<u>Name des Nachrichtenexits</u>	MSGEXIT	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Benutzerdaten des Nachrichtenexits</u>	MSGDATA	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Name des Exits für Nachrichtenwiederholung</u>	MREXIT			Ja	Ja				Ja	
<u>Benutzerdaten des Exits für Nachrichtenwiederholungen</u>	MRDATA			Ja	Ja				Ja	
<u>Zähler für Nachrichtenwiederholungen</u>	MRRTY			Ja	Ja				Ja	
<u>Intervall für Nachrichtenwiederholungen</u>	MRTMR			Ja	Ja				Ja	
<u>Monitoring</u>	MONCHL	Ja	Ja	Ja	Ja		Ja	Ja	Ja	
<u>Netzverbindungspriorität</u>	NETPRTY								Ja	

Tabelle 24. Kanalattribute für die Kanaltypen (Forts.)

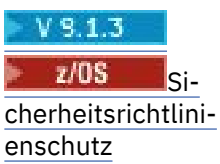

Attributfeld	MQSC-Be- fehlspara- meter	SDR	SVR	RCV R	RQST R	CLNT- CONN	SVR- CONN	CLUS - SDR	CLUS- RCVR	AM QP
<u>Gibt die Ge- schwindigkeit nicht permanenter Nachrichten an.</u>	NPMSPEE D	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Kennwort</u>	PASS- WORD	Ja	Ja		Ja	Ja		Ja		
<u>Port Number</u>	PORT									Ja
<u>Eigenschaften- steuerung</u>	PROPCTL	Ja	Ja					Ja	Ja	
<u>PUT-Berechtigung</u>	PUTAUT			Ja	Ja		Yes (Ja) „2“ auf Seite 100		Ja	
<u>Warteschlangen- managername</u>	QMNAME					Ja				
<u>Name des Emp- fangsexits</u>	RCVEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzerdaten des Empfangs- exits</u>	RCVDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Name des Sicher- heitsexits</u>	SCYEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzerdaten des Sicherheits- exits</u>	SCYDATA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
	SPLPROT	Ja	Ja	Ja	Ja					
<u>Name des Sende- exits</u>	SENDEXIT	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Benutzerdaten des Sendeexits</u>	SENDDA- TA	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Folgenummernse- rie</u>	SEQWRAP	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Gemeinsam ge- nutzte Verbindun- gen</u>	SHA- RECNV					Ja	Ja			
<u>Zähler für Wieder- holungsversuche nach kurzem In- tervall</u>	SHORT- RTY	Ja	Ja					Ja	Ja	

Tabelle 24. Kanalattribute für die Kanaltypen (Forts.)

Attributfeld	MQSC-Befehlsparameter	SDR	SVR	RCV R	RQST R	CLNT-CONN	SVR-CONN	CLUS-SDR	CLUS-RCVR	AM QP
<u>Kurzes Wiederholungsintervall</u>	SHORTTMR	Ja	Ja					Ja	Ja	
<u>SSL-Verschlüsselungsspezifikation</u>	SSLCIPH	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>SSL-Clientauthentifizierung</u>	SSLCAUTH		Ja	Ja	Ja		Ja		Ja	Ja
<u>SSL-Peer</u>	SSLPEER	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<u>Themen-Root</u>	TPROOT									Ja
<u>Name der Übertragungswarteschlange</u>	XMITQ	Ja	Ja							
<u>Transport Type</u>	TRPTYPE	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	
<u>Client-ID verwenden</u>	USECLTID									Ja
<u>Warteschlange für nicht zustellbare Nachrichten verwenden</u>	USEDLQ	Ja	Ja	Ja	Ja			Ja	Ja	
<u>Benutzer-ID</u>	USERID	Ja	Ja		Ja	Ja		Ja		

**Anmerkungen:**

1. Dieses Attribut kann durch keine Verwaltungsschnittstelle abgefragt oder für CLUSSDR-Kanäle festgelegt werden. Sie erhalten eine MQRCCF\_WRONG\_CHANNEL\_TYPE-Nachricht. Das Attribut ist allerdings in CLUSSDR-Kanalobjekten vorhanden (einschließlich MQCD-Strukturen) und kann bei Bedarf durch einen CHAD-Exit programmgesteuert festgelegt werden.
2.  Nur gültig unter z/OS.

**Zugehörige Konzepte**

„Kanalattribute in alphabetischer Reihenfolge“ auf Seite 100

In diesem Abschnitt werden alle Attribute eines Kanalobjekts mit ihren gültigen Werten und ggf. Hinweisen zu ihrer Verwendung beschrieben.

**Zugehörige Verweise**

MQSC-Befehle

**Kanalattribute in alphabetischer Reihenfolge**

In diesem Abschnitt werden alle Attribute eines Kanalobjekts mit ihren gültigen Werten und ggf. Hinweisen zu ihrer Verwendung beschrieben.

IBM MQ implementiert für bestimmte Plattformen unter Umständen nicht alle in diesem Abschnitt aufgeführten Attribute. Auf Ausnahmen und Plattformunterschiede wird ggf. in den einzelnen Attributbeschreibungen hingewiesen.

Das Schlüsselwort, das Sie in MQSC angeben können, wird bei jedem Attribut in eckigen Klammern angegeben.

Die Attribute sind in alphabetischer Reihenfolge geordnet.

## Datum ändern (ALTDATE)

Dieses Attribut ist das Datum, an dem die Definition zuletzt geändert wurde, im Format JJJJ-MM-TT.

Dieses Attribut ist für alle Kanaltypen gültig.

## Zeit ändern (ALTTIME)

Dieses Attribut ist die Zeit, zu der die Definition zuletzt geändert wurde, im Format hh:mm:ss.

Dieses Attribut ist für alle Kanaltypen gültig.

## AMQP keep alive (AMQPKA)

Mit dem Attribut **AMQPKA** können Sie eine Keepalive-Zeit für die AMQP-Clientverbindung angeben. Wenn der AMQP-Client innerhalb des Keepalive-Intervalls keine Frames gesendet hat, wird die Verbindung geschlossen.

Das Attribut **AMQPKA** bestimmt den Wert des Attributs für das Inaktivitätszeitlimit, das von IBM MQ an einen AMQP-Client gesendet wird. Das Attribut gibt einen Zeitraum in Millisekunden an.

Wenn **AMQPKA** auf einen Wert > 0 gesetzt ist, fließt IBM MQ die Hälfte dieses Werts als Attribut für das Inaktivitätszeitlimit. Der Wert 10000 bewirkt beispielsweise, dass der Warteschlangenmanager einen Wert von 5000 für das Inaktivitätszeitlimit sendet. Der Client muss sicherstellen, dass mindestens alle 10.000 Millisekunden Daten an IBM MQ gesendet werden. Wenn IBM MQ in dieser Zeit keine Daten empfängt, geht IBM MQ davon aus, dass der Client seine Verbindung verloren hat, und erzwingt das Schließen der Verbindung mit einer `amqp:resource-limit-exceeded`-Fehlerbedingung.

Der Wert AUTO oder 0 gibt an, dass IBM MQ kein Attribut für das Inaktivitätszeitlimit an den AMQP-Client übergibt.

Ein AMQP-Client kann jedoch einen eigenen Wert für das Inaktivitätszeitlimit übergeben. In diesem Fall übergibt IBM MQ Daten (oder einen leeren AMQP-Frame) mit mindestens dieser Häufigkeit, um den Client darüber zu informieren, dass er verfügbar ist.

## Stapelintervall der Überwachungssignale (BATCHEB)

Mit diesem Attribut können sendende Kanäle vor dem Festschreiben eines Nachrichtenstapels prüfen, ob der empfangende Kanal noch aktiv ist.

Damit ermöglicht das Stapelintervall der Überwachungssignale, dass der Stapel zurückgesetzt wird und nicht unbestätigt bleibt, wenn der empfangende Kanal inaktiv ist. Durch das Zurücksetzen des Stapels bleiben die Nachrichten für die Verarbeitung verfügbar und können zum Beispiel an einen anderen Kanal umgeleitet werden.

Wenn der sendende Kanal vom empfangenden Kanal innerhalb des Überwachungsintervalls für den Stapelbetrieb ein Meldung erhalten hat, wird davon ausgegangen, dass der empfangende Kanal immer noch aktiv ist. Andernfalls wird zur Überprüfung ein Überwachungssignal an den empfangenden Kanal gesendet. Der sendende Kanal wartet die durch das Kanalattribut 'Intervall der Überwachungssignale' (HBINT) angegebene Anzahl an Sekunden auf eine Antwort vom empfangenden Kanal.

Der Wert wird in Millisekunden angegeben und muss im Bereich 0 bis 999999 liegen. Beim Wert 0 werden keine Überwachungssignale für Stapel verwendet.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## Stapelintervall (BATCHINT)

Dieses Attribut gibt an, wie lange (in Millisekunden) der Kanal einen Stapel geöffnet hält, auch wenn sich keine Nachrichten in der Übertragungswarteschlange befinden.

Sie können einen beliebigen Wert von 0 bis 999.999.999 angeben. Der Standardwert ist null.

Wenn Sie kein Stapelintervall angeben, wird der Stapel bei Eintreten einer der folgenden Bedingungen geschlossen:

- Die in BATCHSZ angegebene Anzahl von Nachrichten wurde gesendet.
- Die in BATCHLIM angegebene Anzahl an Byte wurde gesendet.
- Die Übertragungswarteschlange ist leer.

Bei Kanälen mit geringer Auslastung und häufig leerer Übertragungswarteschlange kann die effektive Stapelgröße deutlich unter BATCHSZ liegen.

Mit dem Attribut BATCHINT können Sie die Effizienz der Kanäle steigern, indem Sie die Anzahl kurzer Stapel verringern. Beachten Sie jedoch, dass sich die Reaktionszeit verlängert, weil Stapel länger dauern und Nachrichten länger nicht festgeschrieben werden.

Wenn Sie BATCHINT angeben, werden Stapel nur bei einer der folgenden Bedingungen geschlossen:

- Die in BATCHSZ angegebene Anzahl von Nachrichten wurde gesendet.
- Die in BATCHLIM angegebene Anzahl an Byte wurde gesendet.
- Es gibt keine weiteren Nachrichten in der Übertragungswarteschlange und das Zeitintervall von BATCHINT ist abgelaufen, während auf Nachrichten gewartet wurde (seit Abruf der ersten Nachricht des Stapels).

**Anmerkung:** BATCHINT gibt die gesamte Zeit an, in der auf Nachrichten gewartet wird. Darin ist nicht die Zeit für den Abruf von Nachrichten enthalten, die bereits in der Übertragungswarteschlange verfügbar sind, und nicht die Zeit, die für das Übertragen von Nachrichten benötigt wird.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## Stapel datengrenzwert (BATCHLIM)

Dieses Attribut gibt (in Kilobyte) das Datenvolumen an, das vor dem nächsten Synchronisationspunkt maximal über einen Kanal gesendet werden kann.

Ein Synchronisationspunkt wird erreicht, nachdem die Nachricht, mit der dieser Grenzwert erreicht wurde, vollständig über den Kanal übertragen wurde.

Der Wert muss zwischen 0 und 999999 liegen. Der Standardwert ist 5000.

Der Wert null für dieses Attribut bedeutet, dass es für Stapel auf diesem Kanal keinen Datengrenzwert gibt.

Der Stapel wird beendet, wenn eine der folgenden Bedingungen eintritt:

- BATCHSZ-Nachrichten wurden gesendet.
- BATCHLIM-Bytes wurden gesendet.
- Die Übertragungswarteschlange ist leer und BATCHINT wurde überschritten.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server

- Clustersender
- Clusterempfänger

Dieser Parameter wird auf allen Plattformen unterstützt.

## Stapelgröße (BATCSZ)

Dieses Attribut ist die maximale Anzahl der zu sendenden Nachrichten, bevor ein Synchronisationspunkt erstellt wird.

Die Stapelgröße wirkt sich nicht auf die Nachrichtenübertragungsmethode des Kanals aus. Nachrichten werden immer einzeln übertragen, aber als Stapel festgeschrieben oder zurückgesetzt.

Zur Leistungssteigerung können Sie eine Stapelgröße einstellen, die die maximale Anzahl der zwischen zwei *Synchronisationspunkten* zu übertragenden Nachrichten definiert. Die zu verwendende Stapelgröße wird beim Start eines Kanals vereinbart. Die kleinere der beiden Kanaldefinitionen wird verwendet. Bei einigen Implementierungen wird die Stapelgröße aus der kleinsten der beiden Kanaldefinitionen und den beiden Warteschlangenmanager-MAXUMSGS-Werten berechnet. Die tatsächliche Größe eines Stapels kann geringer ausfallen. Ein Stapel wird beispielsweise beendet, wenn es keine Nachrichten mehr in der Übertragungswarteschlange gibt oder das Stapelintervall abläuft.

Ein großer Wert für die Stapelgröße erhöht den Durchsatz, verlängert aber auch die Wiederherstellungszeiten, weil mehr Nachrichten zurückgesetzt und erneut gesendet werden müssen. Der Standardwert für BATCSZ beträgt 50. Probieren Sie zuerst diesen Wert aus. Sie können einen niedrigeren Wert für BATCSZ wählen, wenn die Kommunikation störanfällig ist und damit die Wahrscheinlichkeit von Wiederherstellungen steigt.

Das Synchronisationspunktverfahren erfordert, dass beim Erstellen eines Synchronisationspunkts eine eindeutige ID der logischen Arbeitseinheit über die Verbindung ausgetauscht wird, um die Stapelfestschreibungsprozeduren zu koordinieren.

Wenn die synchronisierte Stapelfestschreibungsprozedur unterbrochen wird, kann eine *unbestätigte* Situation entstehen. Unbestätigte Situationen werden beim Starten eines Nachrichtenkanals automatisch aufgelöst. Wenn diese Auflösung nicht erfolgreich ist, kann ein manueller Eingriff mit dem Befehl RESOLVE erforderlich sein.

Einige Überlegungen zur Wahl der Stapelgröße:

- Wenn der Wert zu groß ist, wird auf beiden Seiten der Verbindung zu viel Warteschlangenspeicherplatz belegt. Nachrichten belegen Warteschlangenspeicherplatz, wenn sie nicht festgeschrieben werden, und können erst aus den Warteschlangen entfernt werden, wenn sie festgeschrieben wurden.
- Wenn ein kontinuierlicher Nachrichtenfluss zu erwarten ist, können Sie die Leistung eines Kanals durch Erhöhen der Stapelgröße verbessern, da dadurch weniger Bestätigungsflüsse für die Übertragung derselben Anzahl von Bytes erforderlich sind.
- Wenn die Merkmale des Nachrichtenflusses anzeigen, dass Nachrichten unregelmäßig eintreffen, kann die Stapelgröße 1 mit einem relativ großen Unterbrechungsintervall eine bessere Leistung erzielen.
- Der Wert kann im Bereich 1 bis 9999 liegen.
- Auch wenn nicht persistente Nachrichten auf einem schnellen Kanal nicht auf einen Synchronisationspunkt warten, werden sie in die Stapelgröße eingerechnet.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## Zertifikatsbezeichnung (CERTLABL)

Dieses Attribut gibt die Zertifikatsbezeichnung der Kanaldefinition an.

Die Bezeichnung ermittelt, welches persönliche Zertifikat im Schlüsselrepository an den fernen Peer gesendet wird. Das Zertifikat wird wie unter [Digitale Zertifikatsbeschreibungen](#) beschrieben definiert.

Eingehende Kanäle (einschließlich RCVR-, RQSTR-, CLUSRCVR-, nicht qualifizierte SERVER- und SVRCONN-Kanäle) senden das konfigurierte Zertifikat nur, wenn die IBM MQ-Version des fernen Peers die Konfiguration der Zertifikatsbezeichnung vollständig unterstützt und der Kanal ein TLS-CipherSpec verwendet. Ist dies nicht der Fall, bestimmt das Attribut **CERTLABL** des Warteschlangenmanagers das gesendete Zertifikat. Diese Einschränkung besteht, da der Auswahlmechanismus für die Zertifikatsbezeichnung für eingehende Kanäle von einer TLS-Protokollerweiterung abhängt, die nicht in allen Fällen unterstützt wird. Insbesondere Java -Clients und JMS -Clients unterstützen die erforderliche Protokollerweiterung nicht und empfangen nur das vom Warteschlangenmanagerattribut **CERTLABL** konfigurierte Zertifikat, unabhängig von der kanalspezifischen Kennsatzeinstellung.

Ein nicht qualifizierter Serverkanal ist ein Kanal, für den das Feld CONNAME nicht festgelegt wurde.

Dieses Attribut kann durch keine Verwaltungsschnittstelle abgefragt oder für CLUSSDR-Kanäle festgelegt werden. Sie erhalten eine MQRCCF\_WRONG\_CHANNEL\_TYPE-Nachricht. Das Attribut ist allerdings in CLUSSDR-Kanalobjekten vorhanden (einschließlich MQCD-Strukturen) und kann bei Bedarf durch einen CHAD-Exit programmgesteuert festgelegt werden.

Weitere Informationen zu den möglichen Inhalten von Zertifikatsbezeichnungen finden Sie im Abschnitt [Digitale Zertifikatsbezeichnungen - Anforderungen](#).

Dieses Attribut ist für alle Kanaltypen gültig.

**Anmerkung:** Für SSL/TLS muss in der QMGR-Definition ein CERTLABL festgelegt sein. Optional können Sie das CERTLABL auch in der CHANNEL-Definition festlegen.

Das CERTLABL des Warteschlangenmanagers wird überprüft. Es muss ein gültiges persönliches Zertifikat sein, selbst wenn in der CHANNEL-Definition ein CERTLABL festgelegt ist.

## Kanalname (CHANNEL)

Dieses Attribut gibt den Namen der Kanaldefinition an.

Der Name kann bis zu 20 Zeichen lang sein. Da jedoch beide Seiten eines Nachrichtenkanals denselben Namen haben müssen und bei anderen Implementierungen Größenbeschränkungen gelten können, muss die tatsächliche Zeichenzahl unter Umständen darunter liegen.

Kanalnamen dürfen auf zwei Warteschlangenmanagern in einem Netz miteinander verbundener Warteschlangenmanager nie doppelt vorkommen.

Der Name muss aus Zeichen der folgenden Liste bestehen:

Alphabetisch	(A-Z, a-z; Groß-/Kleinschreibung wird beachtet)
Ziffern	(0-9)
Punkt	(.)
Schrägstrich	(/)
Unterstrich	(_)
Prozentzeichen	(%)

### Anmerkung:

1. Eingebettete Leerzeichen sind nicht zulässig. Führende Leerzeichen werden ignoriert.
2. Auf Systemen, die EBCDIC Katakana verwenden, können keine Kleinbuchstaben verwendet werden.

Dieses Attribut ist für alle Kanaltypen gültig.



## Kanalstatistik (STATCHL)

Dieses Attribut steuert die Erfassung von statistischen Daten für Kanäle.

Folgende Werte sind möglich:

### QMGR

Die Erfassung statistischer Daten für diesen Kanal basiert auf der Einstellung des Warteschlangenmanagerattributs STATCHL. Dies ist der Standardwert.

### OFF

Die Erfassung statistischer Daten für diesen Kanal wird inaktiviert.

### LOW

Die Erfassung statistischer Daten für diesen Kanal wird mit einer niedrigen Datenerfassungsrate aktiviert.


### MITTEL

Die Erfassung statistischer Daten für diesen Kanal wird mit einer mittleren Datenerfassungsrate aktiviert.

### HIGH

Die Erfassung statistischer Daten für diesen Kanal wird mit einer hohen Datenerfassungsrate aktiviert.

Weitere Informationen zur Kanalstatistik finden Sie im Abschnitt [Referenzinformationen zur Überwachung](#).

 Auf z/OS -Systemen wird durch die Aktivierung dieses Parameters einfach die Erfassung statistischer Daten aktiviert, unabhängig vom ausgewählten Wert. Die Angabe von LOW, MEDIUM oder HIGH hat keine Auswirkung auf die Ergebnisse. Dieser Parameter muss aktiviert sein, damit Datensätze zur Kanalabrechnung erfasst werden können.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## Kanaltyp (CHLTYPE)

Dieses Attribut gibt den Typ des definierten Kanals an.

Mögliche Kanaltypen:

### Nachrichtentypen:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

### MQI-Kanaltypen:

- Clientverbindung (nur Windows und UNIX)

**Anmerkung:** Sie können Clientverbindungskanäle unter z/OS auch für die Verwendung auf anderen Plattformen definieren.

- Serververbindung

- AMQP

Beide Kanalseiten müssen denselben Namen und kompatible Typen haben:

- Sender mit Empfänger
- Requester mit Server
- Requester mit Sender (für Callback)
- Server mit Empfänger (Server wird als Sender verwendet)
- Clientverbindung mit Serververbindung
- Clustersender mit Clusterempfänger
- AMQP mit AMQP

## Clientkanalgewichtung (CLNTWGHT)

Dieses Attribut gibt eine Gewichtung an, durch die gesteuert wird, welche Clientverbindungskanaldefinition verwendet wird.

Mit dem Attribut CLNTWGHT können Clientkanaldefinitionen auf der Basis ihrer Gewichtung ausgewählt werden, wenn mehrere geeignete Definitionen zur Verfügung stehen.

Wenn ein Client MQCONN ausgibt, um eine Verbindung zu einer Warteschlangenmanagergruppe anzufordern, und dabei einen mit einem Stern beginnenden Warteschlangenmanagernamen angibt, wodurch die Clientgewichtung für mehrere Warteschlangenmanager ermöglicht wird, wird die zu verwendende Definition auf der Basis der Gewichtung ausgewählt, wenn die Definitionstabelle für Clientkanäle (CCDT - Client Channel Definition Table) mehrere geeignete Kanaldefinitionen enthält. Dabei werden gültige Definitionen des Typs CLNTWGHT(0) in alphabetischer Reihenfolge zuerst ausgewählt.

**Anmerkung:** **V9.1.2** Wenn eine JSON-CCDT verwendet wird, können mehrere Kanäle denselben Namen haben. Wenn mehrere Kanäle mit demselben Namen vorhanden sind und den Definitionstyp CLNTWGHT (0) aufweisen, werden die Kanäle in der Reihenfolge ausgewählt, in der sie in der JSON-CCDT definiert sind.

Geben Sie einen Wert im Bereich von 0 bis 99 an. Der Standardwert ist 0.

Der Wert 0 gibt an, dass kein Lastausgleich erfolgt und gültige Definitionen in alphabetischer Reihenfolge ausgewählt werden. Wenn der Lastausgleich aktiviert werden soll, wählen Sie einen Wert im Bereich von 1 bis 99 aus, wobei 1 der niedrigsten und 99 der höchsten Gewichtung entspricht. Die Aufteilung der Verbindungen zwischen zwei oder mehreren Kanälen mit einer Gewichtung ungleich null erfolgt in etwa proportional zum Verhältnis dieser Gewichtungen. Es werden beispielsweise drei Kanäle mit den CLNTWGHT-Werten 2, 4 und 14 zu rund 10 %, 20 % und 70 % der Zeit ausgewählt. Diese Verteilung ist nicht garantiert. Wenn das Attribut AFFINITY der Verbindung auf PREFERRED gesetzt ist, wird für die erste Verbindung eine Kanaldefinition entsprechend den Clientgewichtungen ausgewählt und für die nachfolgenden Verbindungen wird dieselbe Kanaldefinition verwendet.

Dieses Attribut ist nur für den Kanaltyp Clientverbindungskanal gültig.

## Cluster (CLUSTER)

Dieses Attribut ist der Name des Clusters, dem der Kanal zugeordnet ist.

Die maximale Länge beträgt 48 Zeichen gemäß den Regeln für die Benennung von IBM MQ-Objekten.

Maximal einer der resultierenden Werte von CLUSTER und CLUSNL kann andere Zeichen als nur Leerzeichen enthalten. Wenn einer der Werte andere Zeichen als nur Leerzeichen enthält, darf der andere Wert nur aus Leerzeichen bestehen.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

## Clusternamensliste (CLUSNL)

Dieses Attribut ist der Name der Namensliste mit den Clustern, zu denen der Kanal gehört.

Maximal einer der resultierenden Werte von CLUSTER und CLUSNL kann andere Zeichen als nur Leerzeichen enthalten. Wenn einer der Werte andere Zeichen als nur Leerzeichen enthält, darf der andere Wert nur aus Leerzeichen bestehen.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clustersender
- Clusterempfänger

## Kanalattribut CLWLPRTY

Das Kanalattribut CLWLPRTY legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Kanalattribut CLWLPRTY legen Sie die Reihenfolge der Priorität der verfügbaren Clusterziele fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Nachrichten gehen an den Warteschlangenmanager mit dem Kanal mit der höchsten Priorität. Ist dieser Kanal nicht verfügbar, gehen die Nachrichten an den Warteschlangenmanager mit der nächsthöchsten Priorität. Warteschlangenmanager mit niedrigerer Priorität fungieren als Reserve.

Vor der Priorisierung der Kanäle prüft IBM MQ den Kanalstatus. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

### Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Verfügbarkeit eines fernen Warteschlangenmanagers basiert auf dem Status des Kanals für diesen Warteschlangenmanager. Wenn Kanäle gestartet werden, ändert sich ihr Status mehrmals, wobei einige Status für den Algorithmus für das Clusterauslastungsmanagement weniger günstig sind. In der Praxis bedeutet dies, dass Ziele mit einer niedrigeren Priorität (Sicherung) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.
- Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherungsziel gesendet werden, verwenden Sie CLWLPRTY nicht. Ziehen Sie die Verwendung separater Warteschlangen in Betracht, oder CLWLRANK mit einem manuellen Umschalten von der primären auf die Sicherung.

## Kanalattribut CLWLRANK

Das Kanalattribut **CLWLRANK** gibt die Ebene der Kanäle für die Verteilung der Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Verwenden Sie das Kanalattribut **CLWLRANK**, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Sie steuern die Auswahl des Zielorts, indem Sie den Rang der Kanäle festlegen, die einen Warteschlangenmanager am Schnittpunkt der Cluster mit den Gateway-Warteschlangenmanagern verbinden.

Wenn **CLWLRANK** gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen. Ein Beispiel: Nachrichten kommen an einem Gateway-Warteschlangenmanager an, der sie an einen der beiden Warteschlangenmanager weiterleiten kann, die Kanäle mit dem Rang 1 und 2 verwenden. Die Nachrichten werden automatisch an den Warteschlangenmanager gesendet, der durch einen Kanal mit dem höchsten Rang verbunden ist; in diesem Fall ist dies der Warteschlangenmanagerkanal mit dem Rang 2.

IBM MQ ruft den Rang von Kanälen noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Kanäle zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

#### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Würden Sie hingegen auch das Prioritätsattribut **CLWLPRTY** verwenden, würde IBM MQ nur zwischen verfügbaren Zielen auswählen. Steht ein Kanal zum Ziel mit dem höchsten Rang nicht zur Verfügung, wird die Nachricht in der Übertragungswarteschlange zurückgehalten. Erst bei Verfügbarkeit des Kanals wird sie freigegeben. Die Nachricht wird also nicht an das nächste verfügbare Ziel der Rangordnung gesendet.

### **Kanalattribut CLWLWGHT**

Das Kanalattribut CLWLWGHT gibt die Gewichtung von CLUSSDR- und CLUSRCVR-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

Verwenden Sie CLWLWGHT, um mehr Nachrichten an Server mit einer größeren Verarbeitungskapazität zu senden. Je stärker ein Kanal gewichtet ist, desto mehr Nachrichten werden über diesen Kanal versendet.

#### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Standardeinstellung des Parameters CLWLWGHT ist 50. Wird diese Einstellung für einen Kanal geändert, so wird der Lastausgleich abhängig von der Häufigkeit, die die einzelnen Kanäle für eine an eine Clusterwarteschlange gesendete Nachricht ausgewählt wurden. Weitere Informationen finden Sie unter „Algorithmus für das Clusterauslastungsmanagement“ auf Seite 155.

### **Verbindungsaffinität (AFFINITY)**

Dieses Attribut gibt an, ob Clientanwendungen, bei denen mehrfach Verbindungen mit dem gleichen Warteschlangenmanagernamen hergestellt werden, denselben Clientkanal verwenden.

Verwenden Sie dieses Attribut (MQIACH\_CONNECTION\_AFFINITY), wenn mehrere gültige Kanaldefinitionen verfügbar sind.

Folgende Werte sind möglich:

#### **PREFERRED**

Die erste Verbindung eines Prozesses, der eine Definitionstabelle für Clientkanäle (CCDT) liest, erstellt basierend auf der Clientkanalgewichtung eine Liste gültiger Definitionen, in der die Definitionen mit der Gewichtung 0 jeweils in alphabetischer Reihenfolge zuerst aufgeführt sind. Bei jeder Verbindung des Prozesses wird versucht, die Verbindung über die erste Definition der Liste herzustellen. Wenn eine Verbindung nicht erfolgreich ist, wird die nächste Definition verwendet. Nicht erfolgreiche Definitionen mit Clientkanalgewichtungswerten ungleich null werden an das Ende der Liste verschoben. Definitionen mit einer Clientkanalgewichtung von 0 verbleiben am Anfang der Liste und werden für jede Verbindung zuerst ausgewählt.

Jeder Clientprozess mit demselben Hostnamen erstellt immer dieselbe Liste.

Bei Clientanwendungen, die in C, C++ oder im .NET-Programmierframework geschrieben wurden (einschließlich vollständig verwalteter .NET-Clientanwendungen) und für Anwendungen, die IBM MQ classes for Java und IBM MQ classes for JMS verwenden, wird die Liste aktualisiert, wenn die Definitionstabelle für Clientkanäle (CCDT) seit der Erstellung der Liste geändert wurde.

Dieser Wert ist der Standardwert und hat den Wert 1.

## KEINE

Die erste Verbindung eines Prozesses, die eine CCDT liest, erstellt eine Liste gültiger Definitionen. Alle Verbindungen eines Prozesses wählen eine gültige Definition basierend auf der Clientkanalgewichtung aus, wobei Definitionen mit der Gewichtung 0 in alphabetischer Reihenfolge zuerst ausgewählt werden.

Bei Clientanwendungen, die in C, C++ oder im .NET-Programmierframework geschrieben wurden (einschließlich vollständig verwalteter .NET-Clientanwendungen) und für Anwendungen, die IBM MQ classes for Java und IBM MQ classes for JMS verwenden, wird die Liste aktualisiert, wenn die Definitionstabelle für Clientkanäle (CCDT) seit der Erstellung der Liste geändert wurde.

Dieses Attribut ist nur für den Kanaltyp Clientverbindungskanal gültig.

## Verbindungsname (CONNNAME)

Dieses Attribut ist die Kommunikationsverbindungs-ID. Es gibt die Kommunikationsverbindungen an, die von diesem Kanal verwendet werden sollen.

Es ist optional für Serverkanäle, sofern der Serverkanal nicht ausgelöst wird. In diesem Fall muss es einen Verbindungsnamen angeben.

Geben Sie **CONNNAME** als durch Kommas getrennte Liste mit Namen von Maschinen für die angegebene **TRPTYPE** an. In der Regel ist nur ein Systemname erforderlich. Sie können mehrere Systemnamen angeben, um mehrere Verbindungen mit denselben Eigenschaften zu konfigurieren. Die Verbindungen werden normalerweise in der Reihenfolge getestet, in der sie in der Verbindungsliste angegeben sind, bis eine Verbindung erfolgreich eingerichtet werden konnte. Die Reihenfolge wird für Clients geändert, wenn das Attribut **CLNTWGH** angegeben wird. Falls keine Verbindung hergestellt werden kann, versucht der Kanal, wie durch die Kanalattribute festgelegt, die Verarbeitung erneut. Bei Clientkanälen stellen Verbindungslisten eine Alternative zur Konfiguration mehrerer Verbindungen mithilfe von Warteschlangenmanagergruppen dar. Bei Nachrichtenkanälen wird eine Verbindungsliste zur Konfiguration von Verbindungen mit den Alternativadressen eines Multi-Instanz-Warteschlangenmanagers verwendet.

Die Bereitstellung von Namen für Mehrfachverbindungen in einer Liste wurde zuerst in IBM WebSphere MQ 7.0.1 unterstützt. Die Syntax des Parameters **CONNNAME** wird geändert. Frühere Clients und Warteschlangenmanager stellen Verbindungen unter Verwendung des ersten Verbindungsnamens in der Liste her und lesen den Rest der Verbindungsnamen in der Liste nicht. Damit die früheren Clients und Warteschlangenmanager die neue Syntax analysieren können, müssen Sie für den ersten Verbindungsnamen in der Liste eine Portnummer angeben. Durch die Angabe einer Portnummer werden Probleme bei der Verbindungsherstellung von einem Client oder Warteschlangenmanager, der unter einer früheren Version als IBM WebSphere MQ 7.0.1 ausgeführt wird, mit einem Kanal vermieden.

**Multi** Unter Multiplatforms ist die Angabe des TCP/IP-Verbindungsnamensparameters eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM MQ automatisch einen Verbindungsnamen, wobei der Standardport vorausgesetzt und die aktuelle IP-Adresse des Systems verwendet wird. Sie können die Standardportnummer überschreiben, aber die aktuelle IP-Adresse des System weiter verwenden. Lassen Sie für jeden Verbindungsnamen den IP-Namen leer und übergeben Sie die Portnummer in runden Klammern; Beispiel:

```
(1415)
```

Die generierte **CONNNAME** wird immer in der Schreibweise mit Trennzeichen (IPv4) oder im Hexadezimalformat (IPv6) und nicht in Form eines alphanumerischen DNS-Hostnamens generiert.

Die maximale Länge des Namens ist von der Plattform abhängig:

- **Multi** 264 Zeichen.
- **z/OS** 48 Zeichen (siehe Anmerkung 1).

## Transporttyp TCP

CONNNAME ist der Hostname oder die Netzadresse der fernen Maschine (oder der lokalen Maschine bei Clusterempfängerkanälen). Beispiel: (ABC.EXAMPLE.COM), (2001:DB8:0:0:0:0:0:0) oder (127.0.0.1). Sie kann die Portnummer enthalten, z. B. (MACHINE(123)).

**z/OS** Er kann den IP-Namen (IP\_name) einer dynamischen DNS-Gruppe oder einen Network Dispatcher-Eingabeport enthalten.

Wenn Sie eine IPv6-Adresse in einem Netz verwenden, das nur IPv4 unterstützt, wird der Verbindungsname nicht aufgelöst. In einem Netz, das IPv4 und IPv6 verwendet, bestimmt der Verbindungsname zusammen mit der lokalen Adresse, welcher IP-Stack verwendet wird. Weitere Informationen finden Sie im Abschnitt „Lokale Adresse (LOCLADDR)“ auf Seite 116.

## Transporttyp LU 6.2

**Windows** **IBM i** **UNIX** Wenn TPNAME und MODENAME angegeben sind, verwenden Sie den vollständig qualifizierten Namen der Partner-LU.

**Multi** Wenn TPNAME und MODENAME leer sind, geben Sie den Namen des CPI-C-Nebeninformationsobjekts für die bestimmte Plattform an.

**z/OS** Der Wert kann in zwei Formen angegeben werden:

- Name der logischen Einheit

Angaben zur logischen Einheit für den Warteschlangenmanager; diese setzen sich aus dem Namen der logischen Einheit, dem TP-Namen sowie (optional) dem Modusnamen zusammen. Dieser Name kann in einem von drei Formaten angegeben werden:

Format	Beispiel
LU-Name	IGY12355
LU-Name/TP-Name	IGY12345/APING
LU-Name/TP-Name/Modusname	IGY12345/APINGD/#INTER

Beim ersten Format müssen der Name des Transaktionsprogramms und der Modusname mit den Attributen TPNAME und MODENAME angegeben werden. Bei den beiden anderen Formaten müssen diese Attribute leer sein. Für Clientverbindungskanäle ist nur das erste Format erlaubt.

- Symbolischer Name

Symbolischer Bestimmungsname für die Angaben zur logischen Einheit für den Warteschlangenmanager, wie im Datensatz mit den Nebeninformationen definiert. Die Attribute TPNAME und MODENAME müssen leer sein. Beachten Sie, dass sich bei Clusterempfängerkanälen die Nebeninformationen in den anderen Warteschlangenmanagern des Clusters befinden. In diesem Fall kann es sich auch um einen Namen handeln, den ein Exit für die automatische Kanaldefinition in die entsprechenden LU-Informationen für den lokalen Warteschlangenmanager auflösen kann.

Der angegebene oder implizierte Name der logischen Einheit (LU) kann der LU-Name einer generischen VTAM-Ressourcengruppe sein.

## Übertragungsprotokoll NetBIOS

CONNNAME ist der in der fernen Maschine definierte NetBIOS-Name.

## Übertragungsprotokoll SPX

CONNNAME ist eine SPX-artige Adresse aus einer 4-Byte-Netzadresse, einer 6-Byte-Knotenadresse und einer 2-Byte-Socket-Nummer. Geben Sie diese Werte im Hexadezimalformat ein. Trennen Sie die Netz- und Knotenadresse durch einen Punkt, und setzen Sie die Socket-Nummer in eckige Klammern. Beispiel:

```
CONNNAME('0a0b0c0d.804abcde23a1(5e86)')
```

Wenn Sie die Socket-Nummer übergehen, wird die SPX-Standardsocketnummer von IBM MQ verwendet. Der Standardwert ist X'5E86'.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Es ist optional für Serverkanäle, sofern der Serverkanal nicht ausgelöst wird. In diesem Fall muss es einen Verbindungsnamen angeben.

**Anmerkung:**

1. Bei der Länge des Namens können Sie die Begrenzung auf 48 Zeichen auf eine der folgenden Arten umgehen:
  - Richten Sie die DNS-Server z. B. so ein, dass Sie als Hostnamen "myserver" anstelle von "myserver.location.company.com" verwenden können.
  - Verwenden Sie IP-Adressen.
2. Die Definition des Übertragungsprotokolls finden Sie im Abschnitt „[Transporttyp \(TRPTYPE\)](#)“ auf Seite [135](#).

## Nachricht umwandeln (CONVERT)

Dieses Attribut gibt an, dass die Nachricht vor der Übertragung in das vom empfangenden System benötigte Format umgewandelt werden muss.

Anwendungsnachrichtendaten werden in der Regel von der empfangenden Anwendung umgewandelt. Wenn sich der ferne Warteschlangenmanager jedoch auf einer Plattform befindet, von der die Datenumwandlung nicht unterstützt wird, verwenden Sie dieses Kanalattribut, um anzugeben, dass die Nachricht **vor** der Übertragung in das Format umgewandelt werden muss, das für das empfangende System erforderlich ist.

Die gültigen Werte sind 'Ja' (yes) und 'Nein' (no). Wenn Sie 'Ja' (yes) angeben, werden die Anwendungsdaten in der Nachricht vor dem Senden umgewandelt, sofern Sie einen Namen der integrierten Formate angegeben haben oder ein Datenumwandlungsexit für ein benutzerdefiniertes Format verfügbar ist (siehe Abschnitt [Datenumwandlungsexits schreiben](#)). Wenn Sie **Nein** angeben, werden die Anwendungsdaten in der Nachricht vor dem Senden nicht umgewandelt.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## Datenkomprimierung (COMPMSG)

Dieses Attribut ist eine Liste mit den Komprimierungsverfahren für Nachrichtendaten, die vom Kanal unterstützt werden.

Bei Sender-, Server-, Clustersender-, Clusterempfänger- und Clientverbindungskanälen werden die angegebenen Werte in der bevorzugten Reihenfolge aufgelistet. Das erste vom fernen Ende des verwendeten Kanals unterstützte Komprimierungsverfahren wird verwendet. Die von den Kanälen unterstützten Komprimierungsverfahren werden an den Nachrichtenexit des sendenden Kanals übergeben, von dem aus das verwendete Komprimierungsverfahren für jede einzelne Nachricht geändert werden kann. Durch die Kom-



primierung werden die Daten geändert, die an den Sende- bzw. Empfangsexit übergeben werden. Details zur Komprimierung des Nachrichtenheaders finden Sie im Abschnitt „[Headerkomprimierung \(COMPHDR\)](#)“ auf Seite 114.

Folgende Werte sind möglich:

**KEINE**

Es werden keine Nachrichtendaten komprimiert. Dies ist der Standardwert.

**RLE**

Nachrichtendaten werden mittels Lauflängencodierung komprimiert.

**ZLIBFAST**

Die Komprimierung der Nachrichtendaten erfolgt unter Verwendung der ZLIB-Komprimierungstechnik. Dabei wird eine kurze Komprimierungszeit bevorzugt.

ZLIBFAST kann optional durch die zEnterprise-Datenkomprimierungsfunktion entlastet werden. Weitere Informationen finden Sie unter [zEDC Express-Funktion](#).

**ZLIBHIGH**

Die Komprimierung der Nachrichtendaten erfolgt unter Verwendung der ZLIB-Komprimierungstechnik. Dabei wird eine hohe Komprimierungsstufe bevorzugt.

**ANY**

Ermöglicht dem Kanal die Unterstützung aller Komprimierungstechniken, die auch der Warteschlangenmanager unterstützt. Wird nur für Empfänger-, Requester- und Serververbindungskanäle unterstützt.

Dieses Attribut ist für alle Kanaltypen gültig.

## Standardmäßige Verbindungswiederherstellung (DEFRECON)

Gibt an, ob eine Clientverbindung automatisch die Verbindung zu einer Clientanwendung wiederherstellt, wenn die Verbindung unterbrochen wird.

Folgende Werte sind möglich:

**NO (Standardwert)**

Sofern nicht von **MQCONNX** überschrieben, wird die Clientverbindung nicht automatisch wiederhergestellt.

**YES**

Wenn nicht durch **MQCONNX** überschrieben, stellt der Client die Verbindung automatisch wieder her.

**QMGR**

Wenn nicht durch **MQCONNX** überschrieben, stellt der Client die Verbindung automatisch wieder her, aber nur mit demselben Warteschlangenmanager. Die Option QMGR hat dieselbe Wirkung wie MQCNO\_RECONNECT\_Q\_MGR.

**Inaktiviert**

Die Verbindungswiederholung ist inaktiviert, auch wenn sie vom Clientprogramm mit dem MQI-Aufruf **MQCONNX** angefordert wird.

Dieses Attribut ist nur für Clientverbindungskanäle gültig.

## Beschreibung (DESCR)

Dieses Attribut beschreibt die Kanaldefinition und enthält bis zu 64 Byte Text.

**Anmerkung:** Die maximale Anzahl von Zeichen nimmt ab, wenn das System einen Doppelbytezeichensatz verwendet.

Stellen Sie mithilfe von Zeichen aus dem Zeichensatz mit der CCSID des Warteschlangenmanagers sicher, dass der Text ordnungsgemäß umgesetzt wird, wenn er an einen anderen Warteschlangenmanager gesendet wird.

Dieses Attribut ist für alle Kanaltypen gültig.



## Unterbrechungsintervall (DISCINT)

Dieses Attribut definiert die Zeit, nach der ein Kanal geschlossen wird, wenn in diesem Zeitraum keine Nachricht mehr eintrifft.

Dieses Attribut ist ein Zeitlimit (in Sekunden) für die Server, Clustersender-, Sender- und Clusterempfängerkanäle. Das Intervall setzt ein, wenn ein Stapel endet, d. h., wenn die Stapelgröße erreicht wird oder das Stapelintervall abläuft und die Übertragungswarteschlange leer ist. Wenn während des angegebenen Zeitintervalls keine Nachrichten in der Übertragungswarteschlange eintreffen, wird der Kanal geschlossen. (Die Zeitangabe ist ein Näherungswert.)

Der Austausch von Steuerdaten zwischen den beiden Kanalseiten im Rahmen des Systemabschlusses beinhaltet eine Angabe über den Grund für die Schließung. Damit wird sichergestellt, dass die entsprechende Seite des Kanals für einen Neustart verfügbar bleibt.

Sie können eine beliebige Zeit (in Sekunden) von 0 bis 999.999 festlegen. Der Wert 0 bedeutet, dass es keine Unterbrechung gibt und unendlich lange gewartet wird.

Bei Serververbindungskanälen, die das TCP-Protokoll verwenden, gibt das Intervall die Zeit in Sekunden an, bis die Verbindung bei Inaktivität des Clients getrennt wird. Wenn eine Serververbindung für diese Dauer keine Kommunikation von ihrem Partnerclient erhalten hat, beendet sie die Verbindung.

Das Inaktivitätsintervall für Serververbindung gilt zwischen IBM MQ -API-Aufrufen von einem Client.

**Anmerkung:** Ein MQGET-Aufruf mit Wartezeit mit potenziell langer Laufzeit wird nicht als Inaktivität klassifiziert und überschreitet daher nie das Zeitlimit, da DISCINT abläuft.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Serververbindung
- Clustersender
- Clusterempfänger

Bei Serververbindungskanälen, die ein anderes Protokoll als TCP verwenden, wird dieses Attribut ignoriert.

**Anmerkung:** Der Wert für das Unterbrechungsintervall wirkt sich auf die Leistung aus.

Ein niedriger Wert (z. B. wenige Sekunden) kann der Systemleistung abträglich sein, weil der Kanal immer wieder gestartet wird. Ein großer Wert (mehr als eine Stunde) kann bedeuten, dass Systemressourcen unnötig lange belegt werden. Sie können auch ein Intervall der Überwachungssignale angeben. Wenn sich keine Nachrichten in der Übertragungswarteschlange befinden, sendet der sendende MCA einen Austausch von Überwachungssignalen an den empfangenden MCA. Dann kann der empfangende MCA den Kanal in den Quiescemodus versetzen, ohne auf den Ablauf des Unterbrechungsintervalls warten zu müssen. Damit diese beiden Werte effektiv zusammenarbeiten können, muss der Wert des Intervalls der Überwachungssignale deutlich niedriger als der Wert des Unterbrechungsintervalls sein.

Der Standardwert für DISCINT beträgt 100 Minuten. Oft ist jedoch ein Wert von wenigen Minuten sinnvoll, um die Leistung nicht zu beeinträchtigen oder um zu verhindern, dass Kanäle über unnötig lange Zeiträume aktiv bleiben. Sofern für Ihre Umgebung geeignet, können Sie diesen Wert ändern, entweder für jeden einzelnen Kanal oder durch Änderung des Werts in den Standardkanaldefinitionen, z. B. SYSTEM.DEF.SENDER.

Weitere Informationen finden Sie im Abschnitt [Kanäle anhalten und stilllegen](#).

## Disposition (QSGDISP)

Dieses Attribut gibt die Disposition des Kanals in einer Gruppe mit gemeinsamer Warteschlange an. Dieser Parameter ist nur unter z/OS gültig.

Folgende Werte sind möglich:

## **QMGR**

Der Kanal wird in der Seitengruppe des Warteschlangenmanagers definiert, der den Befehl ausführt. Dieser Wert stellt den Standardwert dar.

## **GRUPPE**

Der Kanal befindet sich im gemeinsamen Repository. Dieser Wert ist nur in einer Umgebung mit gemeinsamer Warteschlange zulässig. Wenn ein Kanal mit QSGDISP(GROUP) definiert ist, wird der Befehl DEFINE CHANNEL(name) NOREPLACE QSGDISP(COPY) automatisch generiert und an alle aktiven Warteschlangenmanager gesendet, damit diese lokale Kopien in der Seitengruppe 0 erstellen. Bei Warteschlangenmanagern, die nicht aktiv sind oder die der Gruppe mit gemeinsamer Warteschlange zu einem späteren Zeitpunkt beitreten, wird der Befehl beim Starten des Warteschlangenmanagers generiert.

## **KOPIEREN**

Der Kanal wird in der Seitengruppe des Warteschlangenmanagers definiert, der den Befehl ausführt. Dabei wird die Definition vom Kanal QSGDISP(GROUP) mit demselben Namen kopiert. Dieser Wert ist nur in einer Umgebung mit gemeinsamer Warteschlange zulässig.

Dieses Attribut ist für alle Kanaltypen gültig.

## **Headerkomprimierung (COMPHDR)**

Dieses Attribut ist eine Liste mit den Komprimierungsverfahren für Headerdaten, die vom Kanal unterstützt werden.

Für Sender-, Server-, Clustersender-, Clusterempfänger- und Clientverbindungskanäle werden die Werte in der bevorzugten Reihenfolge angegeben. Dabei wird das erste Komprimierungsverfahren verwendet, das vom fernen Ende des verwendeten Kanals unterstützt wird. Die von den Kanälen unterstützten Komprimierungsverfahren werden an den Nachrichtenexit des sendenden Kanals übergeben, von dem aus das verwendete Komprimierungsverfahren für jede einzelne Nachricht geändert werden kann. Durch die Komprimierung werden die Daten geändert, die an den Sende- bzw. Empfangsexit übergeben werden.

Mögliche Werte:

### **KEINE**

Es werden keine Headerdaten komprimiert. Dies ist der Standardwert.

### **SYSTEM**

Headerdaten werden komprimiert.

Dieses Attribut ist für alle Kanaltypen gültig.

## **Intervall der Überwachungssignale (HBINT)**

Dieses Attribut gibt die geschätzte Zeit zwischen zwei Überwachungssignalen an, die der sendende Nachrichtenkanalagent (Message Channel Agent, MCA) überbrücken muss, wenn keine Nachrichten in der Übertragungswarteschlange vorhanden sind.

Der Austausch von Überwachungssignalen hebt die Blockierung des empfangenden Nachrichtenkanalagenten auf, der auf eingehende Nachrichten wartet, bzw. darauf, dass das Unterbrechungsintervall abläuft. Wenn der empfangende MCA entblockt ist, kann er die Kanalverbindung trennen, ohne den Ablauf des Unterbrechungsintervalls abzuwarten. Durch den Austausch von Überwachungssignalen werden auch die für große Nachrichten zugewiesenen Speicherpuffer freigegeben und Warteschlangen geschlossen, die auf der Empfangsseite des Kanals offen gelassen wurden.

Der Wert wird in Sekunden angegeben und muss zwischen 0 und 999999 liegen. Bei Angabe von Null werden keine Überwachungssignale gesendet. Der Standardwert ist 300. Dieser Wert ist nur sinnvoll, wenn er deutlich kleiner ist als das Unterbrechungsintervall.

Bei Anwendungen, die IBM MQ classes for Java, JMS oder .NET-APIs verwenden, wird der HBINT-Wert auf eine der folgenden Arten bestimmt:

- Entweder durch den Wert im SVRCONN-Kanal, der von der Anwendung verwendet wird.

- Oder durch den Wert im CLNTCONN-Kanal, wenn die Anwendung so konfiguriert wurde, dass sie eine CCDT verwendet.

Bei Serververbindungs- und Clientverbindungskanälen können Überwachungssignale unabhängig voneinander sowohl vom Server als auch vom Client gesendet werden. Wenn innerhalb des Überwachungssignalintervalls keine Daten über den Kanal übertragen wurden, sendet der MQI-Agent der Clientverbindung ein Überwachungssignal, das vom MQI-Agenten der Serververbindung durch ein weiteres Überwachungssignal beantwortet wird. Dies geschieht unabhängig vom Status des Kanals, also zum Beispiel unabhängig davon, ob der Kanal während eines API-Aufrufs oder während des Wartens auf Benutzereingaben vom Client inaktiv ist. Zudem kann der MQI-Agent der Serververbindung den Austausch von Überwachungssignalen mit dem Client initialisieren, und zwar auch hier unabhängig vom Status des Kanals. Um zu verhindern, dass der MQI-Agent der Serververbindung und der MQI-Agent der Clientverbindung gleichzeitig an die jeweils andere Seite Überwachungssignale abgeben, wird das Überwachungssignal des Servers erst ausgegeben, wenn über den Kanal für die Länge des Überwachungssignalintervalls plus 5 Sekunden keine Daten mehr übertragen wurden.

Bei Serververbindungs- und Clientverbindungskanälen im Kanalmodus einer früheren Version von IBM WebSphere MQ 7.0 findet der Austausch von Überwachungssignalen nur statt, wenn ein Servernachrichtenkanalagent auf einen MQGET-Befehl mit angegebener WAIT-Option wartet, den er für eine Clientanwendung ausgegeben hat.

Weitere Informationen dazu, wie MQI-Kanäle in beiden Modi arbeiten, finden Sie im Abschnitt [Sharing-Conversations \(MQLONG\)](#).

### Zugehörige Verweise

[CHANNEL DEFINE CHANNEL](#)

[ALTER CHANNEL](#)

## Keepalive-Intervall (KAINT)

Mit diesem Attribut wird ein Zeitlimit für einen Kanal angegeben.

Dieser Wert wird an den Kommunikationsstack übermittelt und gibt das Keepalive-Timing für den Kanal an. Das Attribut ermöglicht die Angabe unterschiedlicher Keepalive-Werte für jeden Kanal.

Sie können das Attribut KAINTE für jeden Kanal einzeln angeben.

**Multi** Auf Multiplatforms können Sie auf den Parameter zugreifen und ihn ändern. Er wird jedoch nur gespeichert und weitergeleitet, eine funktionale Implementierung des Parameters findet nicht statt. Wenn Sie die Funktionalität des Parameters KAINTE benötigen, verwenden Sie den Parameter für das Intervall der Überwachungssignale (HBINT) gemäß Abschnitt [„Intervall der Überwachungssignale \(HBINT\)“](#) auf Seite 114.

Damit dieses Attribut wirksam wird, muss die TCP/IP-Keepalive-Funktion aktiviert sein. Unter z/OS aktivieren Sie Keepalive, indem Sie den MQSC-Befehl ALTER QMGR TCPKEEP(YES) ausgeben. Auf Multiplatforms wird die TCP/IP-Keepalive-Funktion aktiviert, indem der Parameter KEEPALIVE in der TCP-Zeitengruppe der Konfigurationsdatei für verteilte Warteschlangensteuerung (qm.ini) auf YES gesetzt wird (KEEPALIVE=YES), oder sie wird über den IBM MQ Explorer aktiviert. Keepalive muss mithilfe des Datenbestands für die TCP-Profilkonfiguration auch in TCP/IP selbst aktiviert werden.

Der Wert gibt einen Zeitraum (in Sekunden) an und muss im Bereich 0 bis 99999 liegen. Der Keepalive-Intervallwert 0 zeigt an, dass kein kanalspezifisches Keepalive für den Kanal aktiviert ist und nur der systemweite in TCP/IP eingestellte Keepalive-Wert verwendet wird. Sie können KAINTE auch auf AUTO festlegen. (Dies ist der Standardwert.) Dann basiert der Keepalive-Wert wie folgt auf dem vereinbarten Wert des Intervalls der Überwachungssignale (HBINT):

Vereinbarter HBINT-Wert	KAINTE
>0	Vereinbarter HBINT-Wert + 60 Sekunden
0	0

Dieses Attribut ist für alle Kanaltypen gültig.

Der Wert wird für alle Kanäle ignoriert, bei denen TransportType (TRPTYPE) eine andere Einstellung als TCP oder SPX besitzt.

## Lokale Adresse (LOCLADDR)

Dieses Attribut gibt die lokale Kommunikationsadresse für den Kanal an.

**Anmerkung:** AMQP-Kanäle unterstützen nicht dasselbe Format von LOCLADDR wie andere IBM MQ-Kanäle. Weitere Informationen finden Sie unter „[LOCLADDR für AMQP-Kanäle](#)“ auf Seite 118.

## LOCLADDR für alle Kanäle außer AMQP-Kanälen

Dieses Attribut gilt nur, wenn TCP/IP als Transporttyp (TRPTYPE) verwendet wird. Bei allen anderen Transporttypen wird es ignoriert.

Wenn ein LOCLADDR-Wert angegeben wird, verwendet ein Kanal, der gestoppt und anschließend erneut gestartet wird, die in LOCLADDR angegebene TCP/IP-Adresse weiter. In Wiederherstellungsszenarios kann dieses Attribut nützlich sein, wenn der Kanal über eine Firewall kommuniziert. Es behebt Probleme, die dadurch entstehen, dass der Kanal mit der IP-Adresse des TCP/IP-Stacks erneut gestartet wird, mit dem er verbunden ist. LOCLADDR kann einen Kanal auch zwingen, einen IPv4- oder IPv6-Stack in einem Dual Stack-System oder einen Dualmodus-Stack in einem Einzelstack-System zu verwenden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Wenn LOCLADDR eine Netzadresse enthält, muss diese zu einer Netzschnittstelle des Systems gehören, auf dem der Kanal ausgeführt wird. Für die Definition eines Senderkanals im Warteschlangenmanager ALPHA auf den Warteschlangenmanager BETA verwenden Sie beispielsweise den folgenden MSQC-Befehl:

```
DEFINE CHANNEL(TO.BETA) CHLTYPE(SDR) CONNAME(192.0.2.0) XMITQ(BETA) LOCLADDR(192.0.2.1)
```

Die LOCLADDR-Adresse ist die IPv4-Adresse 192.0.2.1. Dieser Senderkanal wird auf dem System des Warteschlangenmanagers ALPHA ausgeführt. Deshalb muss die IPv4-Adresse zu einer der Netzschnittstellen seines Systems gehören.

Der Wert ist die optionale IP-Adresse und der optionale Port bzw. Portbereich für die abgehende TCP/IP-Kommunikation. Diese Informationen werden im folgenden Format angegeben:

```
LOCLADDR([ip-addr][(low-port[,high-port])][,[ip-addr][(low-port[,high-port])]])
```

Die maximale Länge von **LOCLADDR**, einschließlich mehrerer Adressen, beträgt MQ\_LOCAL\_ADD-RESS\_LENGTH.

Wenn Sie **LOCLADDR** nicht angeben, wird automatisch eine lokale Adresse zugeordnet.

Beachten Sie, dass Sie **LOCLADDR** für einen C-Client mithilfe der Definitionstabelle für Clientkanäle (CCDT) definieren können.

Alle Parameter sind optional. Durch Übergehen des ip-addr-Teils der Adresse wird die Konfiguration einer festen Portnummer für eine IP-Firewall aktiviert. Das Übergehen der Portnummer ist hilfreich, um einen bestimmten Netzadapter auszuwählen, ohne eine eindeutige, lokale Portnummer identifizieren zu müssen. Der TCP/IP-Stack generiert eine eindeutige Portnummer.

Geben Sie [, [ip-addr][(low-port[,high-port])]] für jede zusätzliche lokale Adresse mehrmals an. Verwenden Sie mehrere lokale Adressen, wenn Sie eine bestimmte Untergruppe von lokalen Netzadaptern angeben möchten. Sie können auch [, [ip-addr][(low-port[,high-port])]] verwenden, um eine bestimmte lokale Netzadresse auf verschiedenen Servern darzustellen, die Teil einer Multi-Instanz-Warteschlangenmanagerkonfiguration sind.

#### ip-addr

ip-addr (IP-Adresse) wird in einem von drei Formaten angegeben:

##### IPv4-Dezimalschreibweise mit Punkten

Beispiel: 192.0.2.1

##### IPv6-Hexadezimalschreibweise

Beispiel: 2001:DB8:0:0:0:0:0:0

##### Alphanumerisches Hostnamensformat

Beispiel: WWW.EXAMPLE.COM

#### low-port und high-port

low-port (niedrigster\_Port) und high-port (höchster\_Port) sind Portnummern in runden Klammern.

Die folgende Tabelle zeigt, wie der Parameter **LOCLADDR** verwendet werden kann:

LOCLADDR	Bedeutung
9.20.4.98	Kanal wird lokal an diese Adresse gebunden.
9.20.4.98, 9.20.4.99	Kanal wird an beide IP-Adressen gebunden. Bei der Adresse kann es sich um zwei Netzadapter auf einem einzigen Server oder um einen anderen Netzadapter auf zwei verschiedenen Servern in einer Mehrinstanzkonfiguration handeln.
9.20.4.98(1000)	Kanal wird lokal an diese Adresse und an Port 1000 gebunden.
9.20.4.98(1000,2000)	Lokale Kanalbindung an diese Adresse und den Portbereich 1000 bis 2000
(1000)	Kanal wird lokal an Port 1000 gebunden.
(1000,2000)	Kanal wird lokal an einen Port im Bereich von 1000 bis 2000 gebunden.

Beim Start eines Kanals bestimmen die Werte für den Verbindungsnamen (CONNNAME) und die lokale Adresse (LOCLADDR), welcher IP-Stack für die Kommunikation verwendet wird. Der verwendete IP-Stack wird wie folgt bestimmt:

- Wenn für das System nur ein IPv4-Stack konfiguriert ist, wird immer der IPv4-Stack verwendet. Wenn Sie eine lokale Adresse (LOCLADDR) oder einen Verbindungsnamen (CONNNAME) als IPv6-Netzadresse angeben, wird ein Fehler generiert und der Start des Kanals schlägt fehl.

- Wenn für das System nur ein IPv6-Stack konfiguriert ist, wird immer der IPv6-Stack verwendet. Wenn Sie eine lokale Adresse (LOCLADDR) als IPv4-Netzadresse angeben, wird ein Fehler generiert und der Start des Kanals schlägt fehl. Wenn Sie auf Plattformen, die die IPv6-Adresszuordnung unterstützen, einen Verbindungsnamen (CONNAME) als IPv4-Netzadresse angeben, wird die Adresse einer IPv6-Adresse zugeordnet. Beispiel: xxx.xxx.xxx.xxist : :ffff:xxx.xxx.xxx.xxzugeordnet. Für die Verwendung zugeordneter Adressen werden unter Umständen Protokollumsetzungsprogramme benötigt. Vermeiden Sie die Verwendung zugeordneter Adressen, falls möglich.
- Wenn Sie eine lokale Adresse (LOCLADDR) als IP-Adresse für einen Kanal angeben, wird der Stack für diese IP-Adresse verwendet. Wenn Sie die lokale Adresse (LOCLADDR) als Hostname angeben, der in IPv4- und IPv6-Adressen aufgelöst wird, bestimmt der Verbindungsname (CONNAME), welcher der Stacks verwendet wird. Wenn Sie die lokale Adresse (LOCLADDR) und den Verbindungsnamen (CONNAME) als Hostnamen angeben, die in IPv4- und IPv6-Adressen aufgelöst werden, bestimmt das Warteschlangenmanagerattribut IPADDRV den verwendeten Stack.
- Wenn die IPv4- und IPv6-Stacks für das System konfiguriert sind und keine lokale Adresse (LOCLADDR) für einen Kanal angegeben wird, bestimmt der für den Kanal angegebene Verbindungsname (CONNAME), welcher IP-Stack verwendet wird. Wenn Sie den Verbindungsnamen (CONNAME) als Hostname angeben, der in IPv4- und IPv6-Adressen aufgelöst wird, bestimmt das Warteschlangenmanagerattribut IPADDRV den verwendeten Stack.

**Multi** Unter Multiplatforms können Sie einen lokalen Standardwert für die lokale Adresse festlegen, der für alle Senderkanäle verwendet wird, für die keine lokale Adresse definiert ist. Der Standardwert wird definiert, indem die Umgebungsvariable MQ\_LCLADDR vor dem Starten des Warteschlangenmanagers festgelegt wird. Das Format des Werts stimmt mit dem des MQSC-Attributs LOCLADDR überein.

## Lokale Adressen für Clustersenderkanäle

Clustersenderkanäle übernehmen immer die im Zielwarteschlangenmanager definierte Konfiguration des entsprechenden Clusterempfängerkanals. Dies gilt auch dann, wenn ein lokal definierter Clustersenderkanal mit demselben Namen vorhanden ist; in diesem Fall wird die manuelle Definition nur für die Eingangskommunikation verwendet.

Aus diesem Grund sollte man sich nicht auf die im Clusterempfängerkanal definierte lokale Adresse (LOCLADDR) verlassen, da die IP-Adresse wahrscheinlich nicht dem System gehört, auf dem die Clustersender erstellt werden. Aus diesem Grund sollte die lokale Adresse (LOCLADDR) auf dem Clusterempfänger nur verwendet werden, wenn es einen Grund gibt, nur die Ports, nicht jedoch die IP-Adresse für alle potenziellen Clusterabsender zu beschränken, und diese Ports bekanntermaßen auf allen Systemen verfügbar sind, auf denen möglicherweise ein Clustersenderkanal erstellt wird.

Wenn ein Cluster das Attribut LOCLADDR verwenden muss, um Kanäle für die abgehende Kommunikation an eine bestimmte IP-Adresse zu binden, sollte nach Möglichkeit der Exit für die automatische Kanaldefinition oder die standardmäßige lokale Adresse (LOCLADDR) für den Warteschlangenmanager verwendet werden. Bei Verwendung eines Kanalexits wird die Verwendung des Werts für LOCLADDR aus dem Exit in den automatisch definierten CLUSSDR-Kanälen erzwungen.

Bei Verwendung einer lokalen Adresse (LOCLADDR) für Clustersenderkanäle über einen Exit, bei der es sich nicht um einen Standardwert handelt, oder bei Verwendung eines Standardwerts muss in allen entsprechenden manuell definierten Clustersenderkanälen (beispielsweise zu einem Warteschlangenmanager mit vollständigem Repository) ebenfalls der Wert des Attributs LOCLADDR gesetzt sein, damit die Eingangskommunikation über den Kanal möglich ist.

**Anmerkung:** Wenn das Betriebssystem für den in LOCLADDR bereitgestellten Port (oder für alle Ports, falls ein Portbereich angegeben ist) einen Bindungsfehler zurückgibt, wird der Kanal nicht gestartet; das System gibt eine Fehlernachricht aus.

## LOCLADDR für AMQP-Kanäle

AMQP-Kanäle unterstützen ein anderes Format von LOCLADDR als andere IBM MQ-Kanäle:

## LOCLADDR ( *ip-addr* )

LOCLADDR ist die lokale Kommunikationsadresse für den Kanal. Verwenden Sie diesen Parameter, um das Verwenden einer bestimmten IP-Adresse durch den Client zu erzwingen. LOCLADDR ist auch nützlich, um einen Kanal zu zwingen, eine IPv4 -oder IPv6 -Adresse zu verwenden, wenn eine Auswahl verfügbar ist, oder um einen bestimmten Netzadapter auf einem System mit mehreren Netzadaptern zu verwenden.

Die maximale Länge von LOCLADDR ist MQ\_LOCAL\_ADDRESS\_LENGTH.

Wenn Sie LOCLADDR nicht angeben, wird automatisch eine lokale Adresse zugeordnet.

### **ip-addr**

*ip-addr* ist eine einzelne Netzadresse, die in einem von drei Formaten angegeben wird:

#### **IPv4-Dezimalschreibweise mit Punkten**

Beispiel: 192.0.2.1

#### **IPv6-Hexadezimalschreibweise**

Beispiel: 2001:DB8:0:0:0:0:0:0

#### **Alphanumerisches Hostnamensformat**

Beispiel: WWW.EXAMPLE.COM

Bei Angabe einer IP-Adresse wird nur das Adressformat überprüft. Eine Überprüfung der eigentlichen IP-Adresse findet nicht statt.

## **Zugehörige Konzepte**

Mit automatisch definierten Clustersenderkanälen arbeiten

## **Zähler für Wiederholungsversuche nach langem Intervall (LONGRTY)**

Dieses Attribut gibt die maximale Anzahl an Wiederholungsversuchen des Kanals an, seinem Partner eine Sitzung zuzuordnen.

Das Attribut **long retry count** kann zwischen 0 und 999 999 999 gesetzt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Wenn der erste Zuordnungsversuch fehlschlägt, wird die *short retry count* -Zahl verringert und der Kanal wiederholt die verbleibende Anzahl von Versuchen. Wenn er immer noch fehlschlägt, wiederholt er *long retry count* so oft mit einem Intervall von *long retry interval* zwischen den einzelnen Versuchen. Kann die Operation weiterhin nicht ausgeführt werden, wird der Kanal geschlossen. Der Kanal muss anschließend mit einem Befehl neu gestartet werden; er wird nicht automatisch durch den Kanalinitiator gestartet.

Unter z/OS kann ein Kanal keine Wiederholung eingeben, wenn die maximale Anzahl Kanäle (**MAXCHL**) überschritten wurde.

Auf IBM i-, UNIX- und Windows-Systemen wird nur eine Wiederholung versucht, wenn ein Kanalinitiator ausgeführt wird. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

Wenn der Kanalinitiator (unter z/OS) bzw. der Kanal (auf Multiplatforms) während eines Wiederholungsversuchs gestoppt wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* zurückgesetzt, wenn der Kanalinitiator bzw. der Kanal neu gestartet wird oder wenn eine Nachricht erfolgreich über den Senderkanal versendet wird. Wenn jedoch der Kanalinitiator (unter z/OS) bzw. der Warteschlangenmanager (auf Multiplatforms) beendet und erneut gestartet wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche*

nicht zurückgesetzt. Der Kanal behält die Werte des Zählers für Wiederholungsversuche bei, die vor dem Neustart des Warteschlangenmanagers bzw. vor dem Versenden der Nachricht galten.

Auf IBM i-Systemen, UNIXn und Windows-Systemen:

1. Wenn der Status des Kanals von RETRYING zu RUNNING wechselt, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht unverzüglich zurückgesetzt. Die Zähler werden erst zurückgesetzt, wenn die erste Nachricht erfolgreich über den Kanal versendet wird, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., wenn der lokale Kanal die Anzahl der versendeten Nachrichten bestätigt hat.
2. Der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* werden beim Neustart des Kanals zurückgesetzt.

## **Intervall für lange Wiederholungsversuche (LONGTMR)**

Dieses Attribut ist die ungefähre Länge des Zeitintervalls in Sekunden, das der Kanal bis zum erneuten Versuch eines Verbindungsaufbaus abwartet, während der Modus für lange Wiederholungsversuche aktiviert ist.

Das Intervall zwischen den Verbindungsversuchen kann erhöht werden, wenn der Kanal abwarten muss, bis er aktiv ist.

Der Kanal versucht, so oft, wie durch den *Zähler für lange Wiederholungsversuche* für dieses lange Intervall festgelegt, eine Verbindung herzustellen, nachdem er so oft, wie durch den *Zähler für kurze Wiederholungsversuche* festgelegt, mit dem Warteintervall für kurze Wiederholungsversuche eine Verbindung herzustellen.

Dieses Attribut kann auf einen Wert von null bis 999 999 festgelegt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

## **Name des LU 6.2-Modus (MODENAME)**

Dieses Attribut wird bei LU 6.2-Verbindungen verwendet. Es handelt sich um eine zusätzliche Angabe bei den Sitzungsmerkmalen der Verbindung, wenn die Zuordnung einer Kommunikationssitzung erfolgt.

Bei Verwendung der Nebeninformationen für die SNA-Kommunikation ist der Modusname im Nebenobjekt der CPI-C-Kommunikation oder in den APPC-Nebeninformationen definiert. Dieses Attribut muss leer bleiben bzw. auf den SNA-Modusnamen gesetzt sein.

Der Name muss ein bis acht alphanumerische Zeichen lang sein.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

Er gilt nicht für Empfänger- oder Serververbindungskanäle.



## LU 6.2-Transaktionsprogrammname (TPNAME)

Dieses Attribut wird bei LU 6.2-Verbindungen verwendet. Es handelt sich um den Namen bzw. den generischen Namen des Transaktionsprogramms (MCA), das am fernen Ende der Verbindung ausgeführt wird.

Bei Verwendung der Nebeninformationen für die SNA-Kommunikation ist der Transaktionsprogrammname im CPI-C-Nebenobjekt für Kommunikation oder in den APPC-Nebeninformationen definiert, und dieses Attribut muss leer bleiben. Andernfalls wird dieser Name von den Sender- und Requesterkanälen benötigt.

Der Name kann bis zu 64 Zeichen lang sein.

Der Name muss auf den SNA-Transaktionsprogrammnamen festgelegt sein, es sei denn, der CONNAME enthält einen Nebenobjektnamen, dann muss der Name leer bleiben. Der eigentliche Name wird stattdessen dem CPI-C-Nebenobjekt für Kommunikation bzw. der APPC-Datei für Nebeninformationen entnommen.

Diese Informationen werden auf verschiedenen Plattformen auf verschiedene Weisen festgelegt; weitere Informationen zum Einrichten der Kommunikation für Ihre Plattform finden Sie im Abschnitt [Verteilte Steuerung von Warteschlangen konfigurieren](#).

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender
- Clusterempfänger

## Maximale Anzahl Instanzen (MAXINST)

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals oder AMQP-Kanals an, die gestartet werden können.

In den Unterabschnitten finden Sie Informationen darüber, wie das Attribut für die einzelnen Kanaltypen verwendet wird.

### Zugehörige Konzepte

[Grenzwerte für Serververbindungskanäle](#)

### Zugehörige Verweise

[CHANNEL DEFINE CHANNEL](#)

### **Maximale Anzahl von Instanzen bei Verbindungen über Serververbindungskanäle**

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals an, die gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wird der Wert heruntersgesetzt, sodass er unter der Anzahl der momentan aktiven Instanzen des Serververbindungskanals liegt, hat dies keine Auswirkungen auf die aktiven Kanäle. Neue Instanzen können jedoch erst gestartet werden, nachdem die Ausführung bestehender Instanzen beendet wurde.

### **Maximale Anzahl von Instanzen bei Verbindungen über AMQP-Kanäle**

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines AMQP-Kanals an, die gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wenn ein Client versucht, eine Verbindung herzustellen, und die Anzahl der verbundenen Clients den Wert von MAXINST erreicht hat, schließt der Kanal die Verbindung mit einem Schließvorgangsframe. Der Schließvorgangsframe enthält die folgende Nachricht:

```
amqp:resource-limit-exceeded
```

Wenn sich ein Client mit einer bereits verbundenen ID verbindet (er also eine Clientübernahme vornimmt), ist die Übernahme erfolgreich, und zwar unabhängig davon, ob die Anzahl der verbundenen Clients den Wert von MAXINST bereits erreicht hat.

## Maximale Anzahl Instanzen pro Client (MAXINSTC)

Dieses Attribut gibt die maximale Anzahl simultaner Instanzen eines Serververbindungskanals an, die auf einem einzelnen Client gestartet werden können.

Dieses Attribut kann auf einen Wert von 0 bis 999 999 999 gesetzt werden. Null bedeutet, dass auf diesem Kanal keine Clientverbindungen zulässig sind. Der Standardwert ist 999 999 999.

Wird der Wert heruntersetzt, sodass er unter der Anzahl der momentan auf den einzelnen Clients aktiven Instanzen des Serververbindungskanals liegt, hat dies keine Auswirkungen auf die aktiven Kanäle. Neue Instanzen auf diesen Clients können jedoch erst gestartet werden, nachdem die Ausführung bestehender Instanzen beendet wurde.

Dieses Attribut ist nur für Serververbindungskanäle gültig.

### Zugehörige Konzepte

[Grenzwerte für Serververbindungskanäle](#)

### Zugehörige Verweise

[CHANNEL DEFINE CHANNEL](#)

## Maximale Nachrichtenlänge (MAXMSGL)

Dieses Attribut gibt die maximale Länge von Nachrichten an, die über den Kanal übertragen werden können.

**Multi** Auf IBM MQ for IBM i- Systemen, UNIXn und Windows-Systemen geben Sie einen Wert im Bereich zwischen null und der maximalen Nachrichtenlänge für den Warteschlangenmanager ein. Weitere Informationen finden Sie im Abschnitt zum MAXMSGL-Parameter des Befehls ALTER QMGR in [ALTER QMGR](#).

**z/OS** Geben Sie unter IBM MQ for z/OS einen Wert an, der größer-gleich null und kleiner-gleich 104 857 600 Byte ist (d. h. 100 MB).

Da es auf IBM MQ-Systemen unterschiedliche Implementierungsarten auf verschiedenen Plattformen gibt, ist in manchen Anwendungen die für die Nachrichtenverarbeitung verfügbare Länge möglicherweise begrenzt. Dieser Wert muss einer Größe entsprechen, die Ihr System problemlos verarbeiten kann. Beim Kanalstart wird die niedrigere der zwei Zahlen für die beiden Kanälenden übernommen.

Beim Hinzufügen der digitalen Signatur und des Schlüssels zur Nachricht erhöht [Advanced Message Security](#) die Länge der Nachricht.

**Anmerkung:** Sie können für den Kanal eine maximale Nachrichtengröße von 0 verwenden. Dies bedeutet, dass die Größe auf den Maximalwert des lokalen Warteschlangenmanagers gesetzt werden soll.

Dieses Attribut ist für alle Kanaltypen gültig.

## Name des Nachrichtenkanalagenten (MCANAME)

Dieses Attribut ist reserviert und muss bei Angabe auf Leerzeichen gesetzt werden.

Die maximale Länge beträgt 20 Zeichen.

## Typ des Nachrichtenkanalagenten (MCATYPE)

Dieses Attribut kann den Nachrichtenkanalagenten als *Prozess* oder *Thread* definieren.

Unter IBM MQ for z/OS wird er nur für Kanäle vom Typ Clusterempfänger unterstützt.

Vorteile der Ausführung als Prozess:

- Isolation der einzelnen Kanäle für mehr Integrität
- Für einzelne Kanäle spezifische Jobberechtigung
- Kontrolle über Job-Scheduling

Vorteile von Threads:

- Viel geringere Speicherbelegung
- Einfachere Konfiguration durch Eingabe über die Befehlszeile
- Schnellere Ausführung: es kostet weniger Zeit, einen Thread zu starten, als das Betriebssystem anzuweisen, einen Prozess zu starten

Für die Kanaltypen "Sender", "Server" und "Requester" gilt die Standardeinstellung `Prozess`. Für die Kanaltypen "Clustersender" und "Clusterempfänger" gilt die Standardeinstellung `Thread`. Diese Standardeinstellungen können bei der Installation geändert werden.

Bei Angabe von `Prozess` in der Kanaldefinition wird ein `RUNMQCHL`-Prozess gestartet. Bei Angabe von `Thread` wird der MCA auf einem Thread des `AMQRMPPA`-Prozesses bzw. auf einem Thread des `RUNMQCHI`-Prozesses ausgeführt, wenn `MQNOREMPOOL` angegeben ist. Auf der Maschine, die die eingehenden Zuordnungen empfängt, wird der MCA bei Verwendung von `RUNMQLSR` als Thread ausgeführt. Bei Eingabe von **inetd** wird er als Prozess ausgeführt.

Unter IBM MQ for z/OS wird dieses Attribut nur für Kanäle vom Typ Clusterempfänger unterstützt. Auf anderen Plattformen ist es für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clustersender
- Clusterempfänger

## Benutzer-ID des Nachrichtenkanalagenten (MCAUSER)

Dieses Attribut ist die Benutzer-ID (eine Zeichenfolge), die vom MCA zur Autorisierung für den Zugriff auf IBM MQ-Ressourcen verwendet wird.

**Anmerkung:** Alternativ dazu kann eine Benutzer-ID für einen Kanal, unter der dieser ausgeführt werden soll, über die Verwendung von Kanalauthentifizierungsdatensätzen bereitgestellt werden. Über Kanalauthentifizierungsdatensätze können verschiedene Verbindungen denselben Kanal mit unterschiedlichen Berechtigungsnachweisen verwenden. Wenn für einen Kanal sowohl `MCAUSER` gesetzt ist als auch Kanalauthentifizierungsdatensätze verwendet werden, haben die Kanalauthentifizierungsdatensätze Vorrang. Der Parameter `MCAUSER` in der Kanaldefinition wird nur verwendet, wenn der Kanalauthentifizierungsdatensatz `USERSRC(CHANNEL)` verwendet.

Diese Berechtigung beinhaltet (sofern die `PUT`-Berechtigung `DEF` ist) das Stellen der Nachricht in die Zielwarteschlange für Empfänger- oder Requesterkanäle.

Unter IBM MQ for Windows kann die Benutzer-ID mit dem Format `user@domain` domänenqualifiziert werden, wobei `domain` entweder die Windows-Systemdomäne des lokalen Systems oder eine vertrauenswürdige Domäne sein muss.

Wenn dieses Attribut leer ist, verwendet der MCA seine Standard-Benutzer-ID. Weitere Informationen finden Sie in [DEFINE CHANNEL](#).

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Serververbindung
- Clusterempfänger

### **Zugehörige Konzepte**

Kanalauthentifizierungsdatensätze

### **Name des Nachrichtensexits (MSGEXIT)**

Dieses Attribut gibt den Namen des Benutzerexitprogramms an, das vom Kanalnachrichtensexit ausgeführt werden soll.

Dieses Attribut kann aus einer Namensliste von Programmen bestehen, die nacheinander ausgeführt werden sollen. Lassen Sie das Attribut leer, wenn kein Kanalnachrichtexit aktiv ist.

Das Format und die maximale Länge dieses Attributs sind plattformabhängig, wie bei „Name des Empfangsexits (RCVEXIT)“ auf Seite 129.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

### **Benutzerdaten des Nachrichtensexits (MSGDATA)**

Dieses Attribut gibt die Benutzerdaten an, die an die Kanalnachrichtensexits übergeben werden.

Sie können eine Sequenz von Nachrichtensexits ausführen. Die Beschränkungen für die Benutzerdatenlänge und ein Festlegungsbeispiel von MSGDATA für mehrere Exits entsprechen denen für RCVDATA. (siehe „Benutzerdaten des Empfangsexits (RCVDATA)“ auf Seite 130).

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

### **Name des Nachrichtenwiederholungsexits (MREXIT)**

Dieses Attribut gibt den Namen des Benutzerexitprogramms an, das vom Benutzerexit für Nachrichtenwiederholungen ausgeführt werden soll.

Lassen Sie dieses Attribut leer, wenn kein Exitprogramm für Nachrichtenwiederholung aktiv ist.

Das Format und die maximale Länge des Namens sind plattformabhängig, wie bei „Name des Empfangsexits (RCVEXIT)“ auf Seite 129. Es kann jedoch nur ein Exit für Nachrichtenwiederholungen angegeben werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester

- Clusterempfänger

## **Benutzerdaten des Kanalexits für Nachrichtenwiederholungen (MRDATA)**

Dieses Attribut gibt Daten an, die an den Kanalexit für Nachrichtenwiederholungen übergeben werden, wenn dieser aufgerufen wird.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

## **Zähler für Nachrichtenwiederholungen (MRRTY)**

Dieses Attribut gibt die Anzahl der Versuche des Kanals an, die Nachricht erneut zu übermitteln.

Dieses Attribut steuert die Aktion des MCA nur, wenn kein Name für den Exit für Nachrichtenwiederholung angegeben ist. Wenn der Name des Exits nicht leer ist, wird der Wert von MRRTY an den Exit weitergeleitet, die Anzahl der Wiederholungsversuche (sofern überhaupt welche durchgeführt werden) wird dann aber durch den Exit, nicht durch dieses Attribut, bestimmt.

Der Wert muss im Bereich von 0 bis 999 999 999 liegen. Der Wert 0 bedeutet, dass keine weiteren Versuche ausgeführt werden. Der Standardwert ist 10.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

## **Intervall für Nachrichtenwiederholungen (MRTMR)**

Dieses Attribut gibt die Mindestzeitspanne an, die vergehen muss, bevor der Kanal die MQPUT-Operation erneut versuchen kann.

Dieses Zeitintervall wird in Millisekunden angegeben.

Dieses Attribut steuert die Aktion des MCA nur, wenn kein Name für den Exit für Nachrichtenwiederholung angegeben ist. Wenn der Name des Exits nicht leer ist, wird der Wert von MRTMR zur Verwendung durch den Exit an den Exit übergeben, das Wiederholungsintervall wird jedoch durch den Exit gesteuert, nicht durch dieses Attribut.

Der Wert muss im Bereich von 0 bis 999 999 999 liegen. Der Wert 0 bedeutet, dass eine Wiederholung so schnell wie möglich ausgeführt wird (wenn der Wert für MRRTY größer als null ist). Der Standardwert ist 1000.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Clusterempfänger

## **Überwachung (MONCHL)**

Dieses Attribut steuert die Erfassung von Online-Überwachungsdaten.

Mögliche Werte:

### **QMGR**

Die Erfassung der Daten aus der Online-Überwachung wird aus der Einstellung des Attributs MONCHL des Warteschlangenmanagerobjekts übernommen. Dies ist der Standardwert.

**OFF**

Die Erfassung von Onlineüberwachungsdaten wird für diesen Kanal inaktiviert.

**LOW**

Niedrige Datenerfassungsrate mit minimalen Auswirkungen auf die Systemleistung. Die abgebildeten Überwachungsergebnisse sind möglicherweise nicht aktuell.

**MITTEL**

Mittlere Datenerfassungsrate mit begrenzten Auswirkungen auf die Leistung des Systems.

**HIGH**

Hohe Datenerfassungsrate mit möglichen Auswirkungen auf die Leistung des Systems. Die abgebildeten Überwachungsergebnisse sind jedoch sehr aktuell.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Serververbindung
- Clustersender
- Clusterempfänger

Weitere Informationen zu Überwachungsdaten finden Sie im Abschnitt [Überwachungsdaten von Warteschlangen und Kanälen anzeigen](#).

**Kanalattribut NETPRTY**

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Verwenden Sie das Attribut NETPRTY, um ein Netz als primäres Netz und ein weiteres Netz als Sicherungsnetz festzulegen. Bei einer Gruppe von Kanälen mit gleichem Rang wird durch Clustering der Pfad mit der höchsten Priorität ausgewählt, wenn mehrere Pfade zur Verfügung stehen.

Üblicherweise wird das Kanalattribut NETPRTY verwendet, um zwischen Netzen zu unterscheiden, die die gleichen Zielorte verbinden, aber unterschiedliche Aufwände verursachen oder unterschiedliche Geschwindigkeiten aufweisen.

**Anmerkung:** Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).

**Geschwindigkeit für nicht persistente Nachrichten (NPMSPEED)**

Dieses Attribut gibt die Geschwindigkeit an, mit der nicht persistente Nachrichten gesendet werden sollen.

Mögliche Werte:

**NORMAL**

Nicht persistente Nachrichten in einem Kanal werden innerhalb von Transaktionen übertragen.

**FAST**

Nicht persistente Nachrichten in einem Kanal werden nicht innerhalb von Transaktionen übertragen.

Der Standardwert ist FAST. Der Vorteil besteht darin, dass nicht persistente Nachrichten sehr viel schneller zum Abruf verfügbar sind. Der Nachteil ist, dass Nachrichten bei einem Übertragungsfehler oder einem Stopp des Kanals während der Nachrichtenübertragung verloren gehen können, weil sie nicht Teil einer Transaktion sind. Siehe [Nachrichtensicherheit](#).

**Anmerkungen:**

1. Wenn die aktiven Wiederherstellungsprotokolle für IBM MQ for z/OS häufiger als erwartet wechseln und archivieren, da die Nachrichten, die über einen Kanal gesendet werden, nicht persistent sind, kann die Einstellung NPMSPEED (FAST) sowohl auf der sendenden als auch auf der empfangenden Seite des Kanals das SYSTEM.CHANNEL.SYNCQ -Aktualisierungen.
2. Wenn es wegen der Aktualisierungen von SYSTEM.CHANNEL.SYNCQ zu einer hohen CPU-Auslastung kommt, kann das Festlegen von NPMSPEED(FAST) die CPU-Auslastung erheblich verringern.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

### **Kennwort (PASSWORD)**

Dieses Attribut gibt ein Kennwort an, das der MCA beim Herstellungsversuch einer sicheren LU 6.2-Sitzung zu einem fernen MCA verwenden kann.

Sie können ein Kennwort mit maximal 12 Zeichen angeben, von dem jedoch nur die ersten 10 Zeichen verwendet werden.

Es ist für die Kanaltypen Sender, Server, Requester und Clientverbindung gültig.

Unter IBM MQ for z/OS ist dieses Attribut nur für Clientverbindungskanäle gültig. Auf anderen Plattformen ist es für folgende Kanaltypen gültig:

- Sender
- Server
- Requester
- Clientverbindung
- Clustersender

### **Portnummer (PORT)**

Geben Sie die Portnummer an, die für die Verbindung des AMQP-Clients verwendet wird.

Der Standardport für AMQP 1.0-Verbindungen ist 5672. Falls Port 5672 bereits verwendet wird, können Sie einen anderen Port angeben.

### **PUT-Berechtigung (PUTAUT)**

Dieses Attribut gibt den Typ der vom MCA auszuführenden Sicherheitsverarbeitung an.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Empfänger
- Requester
- Serververbindung (nur z/OS)
- Clusterempfänger

Dieses Attribut wird verwendet, um den Typ der Sicherheitsverarbeitung festzulegen, den der MCA bei Ausführung verwenden soll:

- Ein MQPUT-Befehl an die Zielwarteschlange (für Nachrichtenkanäle) oder
- ein MQI-Aufruf (für MQI-Kanäle).

**z/OS** Unter z/OS die markierten Benutzer-IDs, deren Anzahl von der Einstellung für das hlq.RES-LEVEL-Profil der MQADMIN-RACF-Klasse abhängt. Je nachdem, welche Zugangsstufe die Benutzer-ID des Kanalinitiators für das Profil hlq.RESLEVEL hat, sind null, ein oder zwei Benutzer-IDs markiert. Unter [RESLEVEL- und Kanalinitiator-Verbindungen](#) finden Sie Anweisungen zur Einsicht in die markierten Benutzer-IDs. Weitere Informationen zu markierten Benutzer-IDs finden Sie unter [Vom Kanalinitiator verwendete Benutzer-IDs](#).

Sie können eine der folgenden Optionen wählen:

#### **Prozesssicherheit, auch als Standardberechtigung (DEF) bezeichnet**

Die Standard-Benutzer-ID wird verwendet.

Auf allen Plattformen außer z/OS wird zur Überprüfung der Öffnungsberechtigung für die Warteschlange die Benutzer-ID des Prozesses bzw. des Benutzers verwendet, der den MCA am empfangenden Ende des Nachrichtenkanals ausführt.

Unter z/OS wird je nach Anzahl der zu überprüfenden Benutzer-IDs entweder die aus dem Netz erhaltene oder die von [MCAUSER](#) abgeleitete Benutzer-ID verwendet.

Die Warteschlangen werden mit dieser Benutzer-ID und der Öffnungsoption MQOO\_SET\_ALL\_CONTEXT geöffnet.

#### **Kontextsicherheit (CTX)**

Die Benutzer-ID aus den Kontextinformationen der Nachricht wird als alternative Benutzer-ID verwendet.

Die *UserIdentifier* im Nachrichtendeskriptor wird in das Feld *AlternateUserId* im Objektdeskriptor verschoben. Die Warteschlange wird mit den Öffnungsoptionen MQOO\_SET\_ALL\_CONTEXT und MQOO\_ALTERNATE\_USER\_AUTHORITY geöffnet.

Auf allen Plattformen außer z/OS wird zur Überprüfung der Öffnungsberechtigung für die Warteschlange für MQOO\_SET\_ALL\_CONTEXT und MQOO\_ALTERNATE\_USER\_AUTHORITY die Benutzer-ID des Prozesses bzw. des Benutzers verwendet, der den MCA am empfangenden Ende des Nachrichtenkanals ausführt. Die Benutzer-ID, die zum Überprüfen der Öffnungsberechtigung für die Warteschlange für MQOO\_OUTPUT verwendet wird, ist die *UserIdentifier* im Nachrichtendeskriptor.

Unter z/OS kann je nach Anzahl der zu überprüfenden Benutzer-IDs die aus dem Netz erhaltene oder die von [MCAUSER](#) abgeleitete Benutzer-ID verwendet werden, ebenso wie die Benutzer-ID aus den Kontextinformationen des Nachrichtendeskriptors.

Kontextsicherheit (CTX) wird für Serververbindungskanäle nicht unterstützt.

#### **Sicherheit ONLYMCA (Nur Nachrichtenkanalagent)**

Die von [MCAUSER](#) abgeleitete Benutzer-ID wird verwendet.

Die Warteschlangen werden mit der Öffnungsoption MQOO\_SET\_ALL\_CONTEXT geöffnet.

Dieser Wert wird nur unter z/OS unterstützt.

#### **Sicherheit ALTMCA (Alternativer Nachrichtenkanalagent)**

Abhängig von der Anzahl der zu überprüfenden Benutzer-IDs kann die Benutzer-ID aus den Kontextinformationen (Feld *UserIdentifier*) im Nachrichtendeskriptor sowie die von [MCA-Benutzer](#) abgeleitete Benutzer-ID verwendet werden.



Dieser Wert wird nur unter z/OS unterstützt.

Weitere Details zu Kontextfeldern und Öffnungsoptionen finden Sie im Abschnitt [Kontextinformationen steuern](#).

Weitere Informationen zu Sicherheit finden Sie hier:

- [Sicherheit](#)
- **ULW** [Sicherheit unter UNIX, Linux, and Windows einrichten](#)



-  [Sicherheit unter IBM i einrichten](#)
-  [Sicherheit unter z/OS einrichten](#)

## Name des Warteschlangenmanagers (QMNAME)

Dieses Attribut gibt den Namen des Warteschlangenmanagers oder der Warteschlangenmanagergruppe an, zu dem bzw. der von einer IBM MQ MQI client-Anwendung eine Verbindung angefordert werden kann.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Clientverbindung

## Name des Empfangsexits (RCVEXIT)

Dieses Attribut gibt den Namen des Benutzerexitprogramms an, das vom Empfangsbenutzerexit des Kanals ausgeführt werden soll.

Dieses Attribut kann aus einer Namensliste von Programmen bestehen, die nacheinander ausgeführt werden sollen. Ist kein Kanalempfangsbenutzerexit aktiv, wird dieses Attribut leer gelassen.

Das Format und die maximale Länge dieses Attributs hängen von der Plattform ab:

- Unter z/OS handelt es sich um einen Lademodulnamen mit einer maximalen Länge von 8 Zeichen bzw. bei Clientverbindungskanälen mit maximal 128 Zeichen.
- Auf IBM i wird folgendes Format verwendet:

```
libname/progname
```

wenn es in CL-Befehlen verwendet wird.

Bei Angabe in IBM MQ-Befehlen (MQSC) hat es das Format:

```
progname libname
```

Dabei belegt *progname* die ersten 10 Zeichen und *libname* die zweiten 10 Zeichen (beide mit Leerzeichen aufgefüllt, falls erforderlich). Die maximale Länge der Zeichenfolge beträgt 20 Zeichen.

- Auf Windows wird folgendes Format verwendet:

```
dllname(functionname)
```

Dabei wird *dllname* ohne das Suffix .DLL angegeben. Die maximale Länge der Zeichenfolge beträgt 40 Zeichen.

- Auf UNIX wird folgendes Format verwendet:

```
libraryname(functionname)
```

Die maximale Länge der Zeichenfolge beträgt 40 Zeichen.

Bei der automatischen Definition des Clustersenderkanals unter z/OS werden Kanalexitnamen in das z/OS-Format konvertiert. Wenn Sie die Konvertierung von Exitnamen steuern möchten, können Sie einen Exit für die automatische Kanaldefinition eingeben. Weitere Informationen finden Sie unter [Exitprogramm für die automatische Kanaldefinition](#).

Sie können eine Namensliste von Empfangs-, Sende- oder Nachrichtenexitprogrammen angeben. Die Namen müssen durch ein Komma, ein Leerzeichen oder beides getrennt sein. Beispiel:

```
RCVEXIT(exit1 exit2)
MSGEXIT(exit1,exit2)
SENDEXIT(exit1, exit2)
```

Die Gesamtlänge der Zeichenfolge aus Exitnamen und Zeichenfolgen aus Benutzerdaten für einen bestimmten Exittyp ist auf 500 Zeichen begrenzt. In IBM MQ for IBM i können bis zu 10 Exitnamen aufgelistet werden. In IBM MQ for z/OS können bis zu acht Exitnamen aufgelistet werden.

Dieses Attribut ist für alle Kanaltypen gültig.

## Benutzerdaten des Empfangsexits (RCVDATA)

Dieses Attribut gibt die Benutzerdaten an, die an das Empfangsexit übermittelt werden.

Sie können eine Reihe von Empfangsexits ausführen. Die Zeichenfolge aus Benutzerdaten für eine Reihe von Exits muss jeweils durch Komma, Leerzeichen oder beides getrennt werden. Beispiel:

```
RCVDATA(exit1_data exit2_data)
MSGDATA(exit1_data,exit2_data)
SENDDATA(exit1_data, exit2_data)
```

Auf IBM MQ for UNIX- und Windows-Systemen ist die Länge der Zeichenfolge aus Exitnamen und Zeichenfolgen aus Benutzerdaten auf 500 Zeichen begrenzt. In IBM MQ for IBM i können Sie bis zu 10 Exitnamen angeben und die Länge der Benutzerdaten ist jeweils auf 32 Zeichen beschränkt. Unter IBM MQ for z/OS können Sie bis zu acht Zeichenfolgen aus Benutzerdaten angeben, von denen jede 32 Zeichen lang sein kann.

Dieses Attribut ist für alle Kanaltypen gültig.

## Name des Sicherheitsexits (SCYEXIT)

Dieses Attribut gibt den Namen des Exitprogramms an, das vom Kanalsicherheitsexit ausgeführt werden muss.

Ist kein Kanalsicherheitsexit aktiv, wird das Attribut leer gelassen.

Das Format und die maximale Länge des Namens sind plattformabhängig, wie bei „[Name des Empfangsexits \(RCVEXIT\)](#)“ auf Seite 129. Sie können jedoch nur ein Sicherheitsexit angeben.

Dieses Attribut ist für alle Kanaltypen gültig.

## Benutzerdaten des Sicherheitsexits (SCYDATA)

Dieses Attribut gibt die Benutzerdaten an, die an das Sicherheitsexit übergeben werden.

Die maximal zulässige Länge beträgt 32 Zeichen.

Dieses Attribut ist für alle Kanaltypen gültig.

## Name des Sendeexits (SENDEXIT)

Dieses Attribut gibt den Namen des Exitprogramms an, das vom Kanalsendeexit ausgeführt wird.

Dieses Attribut kann eine Liste von Programmnamen sein, die nacheinander ausgeführt werden müssen. Lassen Sie das Attribut leer, wenn kein Sendeexit aktiv ist.

Das Format und die maximale Länge dieses Attributs sind plattformabhängig, wie bei „[Name des Empfangsexits \(RCVEXIT\)](#)“ auf Seite 129.

Dieses Attribut ist für alle Kanaltypen gültig.

## Benutzerdaten des Sendeexits (SENDDATA)

Dieses Attribut gibt die Benutzerdaten an, die an das Sendeexit übergeben werden.

Sie können eine Sequenz von Sendeexits ausführen. Die Beschränkungen für die Benutzerdatenlänge und die Angabe von SENDDATA für mehrere Exits entsprechen denen für RCVDATA. (siehe „[Benutzerdaten des Empfangsexits \(RCVDATA\)](#)“ auf Seite 130).

Dieses Attribut ist für alle Kanaltypen gültig.

## Wert für Folgenummernserie (SEQWRAP)

Dieses Attribut gibt die höchste Nummer an, die die Nachrichtenfolgennummer erreicht, bevor sie wieder bei 1 beginnt.

Der Wert der Nummer muss hoch genug sein, um zu vermeiden, dass eine Nummer erneut verwendet wird, während diese noch durch eine frühere Nachricht in Verwendung ist. Den beiden Kanalenden muss derselbe Wert für die Folgenummernserie zugeordnet sein, wenn der Kanal gestartet wird; andernfalls tritt ein Fehler auf.

Hier kann ein Wert zwischen 100 und 999 999 999 angegeben werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester
- Clustersender
- Clusterempfänger

## Kurzer Wiederholungszähler (SHORTRTY)

Dieses Attribut gibt die maximale Anzahl an Wiederholungsversuchen des Kanals an, seinem Partner eine Sitzung zuzuordnen.

Das Attribut *Zähler für kurze Wiederholungsversuche* kann auf einen Wert zwischen null und 999 999 999 gesetzt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Wenn mehrere IP-Adressen innerhalb des Kanals definiert wurden und eine erneute Verbindung erforderlich ist, wertet IBM MQ die Kanaldefinition aus und versucht, eine Verbindung zu jeder IP-Adresse in der Reihenfolge herzustellen, in der sie definiert ist, bis entweder eine erfolgreiche Verbindung hergestellt wurde oder alle Adressen versucht wurden.

In diesem Fall bezieht sich SHORTRTY auf die Gesamtzahl der Versuche, die der gesamte Kanal versucht, die Verbindung wiederherzustellen, und nicht auf die einzelnen IP-Adressen.

Wenn der erste Zuordnungsversuch fehlschlägt, wird der Wert für *short retry count* verringert und der Kanal wiederholt die verbleibenden Wiederholungen mit einem Intervall, das im Attribut **short retry interval** definiert ist, zwischen den einzelnen Versuchen. Wenn er immer noch fehlschlägt, wiederholt er *long retry count* die Anzahl der Wiederholungen mit einem Intervall von *long retry interval* zwischen den einzelnen Versuchen. Kann die Operation weiterhin nicht ausgeführt werden, wird der Kanal geschlossen.

Unter z/OS kann ein Kanal keine Wiederholung eingeben, wenn die maximale Anzahl Kanäle (**MAXCHL**) überschritten wurde.

Auf IBM i-, UNIX- und Windows-Systemen wird nur eine Wiederholung versucht, wenn ein Kanalinitiator ausgeführt wird. Der Kanalinitiator muss die Initialisierungswarteschlange überwachen, die in der Definition der vom Kanal genutzten Übertragungswarteschlange festgelegt ist.

Wenn der Kanalinitiator (unter z/OS) bzw. der Kanal (auf Multiplatforms) während eines Wiederholungsversuchs gestoppt wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* zurückgesetzt, wenn der Kanalinitiator bzw. der Kanal neu gestartet wird oder wenn eine Nachricht erfolgreich über den Senderkanal versendet wird. Wenn jedoch der Kanalinitiator

(unter z/OS) bzw. der Warteschlangenmanager (auf Multiplatforms) beendet und erneut gestartet wird, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht zurückgesetzt. Der Kanal behält die Werte des Zählers für Wiederholungsversuche bei, die vor dem Neustart des Warteschlangenmanagers bzw. vor dem Versenden der Nachricht galten.

Auf IBM i-Systemen, UNIXn und Windows-Systemen:

1. Wenn der Status des Kanals von RETRYING zu RUNNING wechselt, werden der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* nicht unverzüglich zurückgesetzt. Die Zähler werden erst zurückgesetzt, wenn die erste Nachricht erfolgreich über den Kanal versendet wird, nachdem der Kanal in den Status RUNNING gewechselt ist, d. h., wenn der lokale Kanal die Anzahl der versendeten Nachrichten bestätigt hat.
2. Der *Zähler für kurze Wiederholungsversuche* und der *Zähler für lange Wiederholungsversuche* werden beim Neustart des Kanals zurückgesetzt.

## Intervall für kurze Wiederholungsversuche (SHORTTMR)

Dieses Attribut gibt die ungefähre Länge des Zeitintervalls in Sekunden an, das der Kanal bis zum erneuten Versuch eines Verbindungsaufbaus abwartet, während der Modus für kurze Wiederholungsversuche aktiviert ist.

Das Intervall zwischen zwei Verbindungsversuchen kann größer sein, wenn ein Kanal abwarten muss, bis er aktiv ist.

Dieses Attribut kann auf einen Wert von null bis 999 999 festgelegt werden.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Clustersender
- Clusterempfänger

Wenn mehrere IP-Adressen innerhalb des Kanals definiert wurden und eine erneute Verbindung erforderlich ist, wertet IBM MQ die Kanaldefinition aus und versucht, eine Verbindung zu jeder IP-Adresse in der Reihenfolge herzustellen, in der sie definiert ist, bis entweder eine erfolgreiche Verbindung hergestellt wurde oder alle Adressen versucht wurden.

In diesem Fall bezieht sich SHORTTMR darauf, wie lange der gesamte Kanal auf den Neustart des Verbindungsprozesses wartet, und nicht auf die einzelnen IP-Adressen.

## Sicherheitsrichtlinienschutz (SPLPROT)

Dieses Attribut gibt an, wie ein Server-zu-Server-Nachrichtenkanalagent mit dem Nachrichtenschutz umgehen soll, wenn AMS aktiv ist und eine gültige Richtlinie vorhanden ist.

Dieses Attribut kann auf Folgendes gesetzt werden:

### **PASSTHRU**

Auf Sender-, Server-, Empfänger- und Requesterkanälen

### **REMOVE**

Auf Sender- und Serverkanälen

### **ASPOLICY**

Auf Empfänger- und Requesterkanälen

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server
- Empfänger
- Requester

## SSL-Verschlüsselungsspezifikation (SSLCIPH)

Das Attribut **SSLCIPH** gibt eine einzelne CipherSpec für eine TLS-Verbindung an.

Jede IBM MQ-Kanaldefinition enthält das Attribut **SSLCIPH**. Der Wert besteht aus einer Zeichenfolge mit einer maximalen Länge von 32 Zeichen.

Das Attribut **SSLCIPH** ist nur für Kanäle vom Transporttyp (**TRPTYPE**) TCP-gültig. Ist **TRPTYPE** nicht TCP-, werden die Daten ignoriert und es wird keine Fehlernachricht ausgegeben.

### Anmerkungen:

- Das Attribut **SSLCIPH** kann einen leeren Wert enthalten, d. h., Sie verwenden kein TLS. Wenn ein Ende des Kanals über ein leeres Attribut **SSLCIPH** verfügt, muss auch das andere Ende des Kanals über ein leeres Attribut **SSLCIPH** verfügen.
- Wenn **SSLCIPH** dagegen einen nicht leeren Wert hat, versucht der Kanal, die angegebene Verschlüsselung für die Nutzung von TLS zu verwenden. Auch in diesem Fall müssen beide Enden des Kanals denselben **SSLCIPH**-Wert aufweisen.
- Für die Regel, dass der Wert des Attributs **SSLCIPH** an beiden Enden eines Kanals identisch sein muss, gibt es nur eine Ausnahme: Auf einem vollständig verwalteten .NET-Client kann der Sonderwert \*NEGOTIATE angegeben werden. Diese Option ermöglicht dem Kanal die Auswahl der neuesten Protokollversion, die vom .NET-Framework unterstützt wird sowie das Aushandeln eines CipherSpec-Werts, der vom Server unterstützt wird.

Weitere Informationen zu **SSLCIPH** finden Sie unter [KANAL DEFINIEREN](#) und [CipherSpecs angeben](#).

## SSL-Clientauthentifizierung (SSLCAUTH)

Das Attribut **SSLCAUTH** gibt an, ob der Kanal ein TLS-Zertifikat von einem TLS-Client empfangen und authentifizieren muss.

Das Attribut **SSLCAUTH** ist mit Ausnahme von Senderkanälen für alle Kanaltypen gültig, die einen Kanalinitialisierungsfluss empfangen. Dieses Attribut ist für folgende Kanaltypen gültig:

- Server
- Empfänger
- Requester
- Serververbindung
- Clusterempfänger

**SSLCAUTH** ist ein optionales Attribut. Für dieses Attribut sind die folgenden Werte möglich:

### OPTIONAL

Wenn der TLS-Peer-Client ein Zertifikat sendet, wird das Zertifikat normal verarbeitet; die Authentifizierung schlägt aber nicht fehl, wenn kein Zertifikat gesendet wird.

### ERFORDERLICH

Wenn der TLS-Client kein Zertifikat sendet, schlägt die Authentifizierung fehl.

Der Standardwert ist REQUIRED.

Sie können in einer Nicht-TLS-Kanaldefinition einen Wert für **SSLCAUTH** angeben. Dabei handelt es sich um eine Kanaldefinition, in der das Attribut **SSLCIPH** fehlt oder leer ist.

Weitere Informationen zu **SSLCAUTH** finden Sie in den Abschnitten [DEFINE CHANNEL \(MQTT\)](#) und [Sicherheit](#).

## SSL-Peer (SSLPEER)

Das Attribut **SSLPEER** wird verwendet, um den definierten Namen (DN) des Zertifikats vom Peer-Warteschlangenmanager oder Client am anderen Ende eines IBM MQ-Kanals zu überprüfen.

Das Attribut **SSLPEER** ist für alle Kanaltypen gültig.

**Anmerkung:** Alternativ können zur Beschränkung von Verbindungen auf bestimmte Kanäle durch Überprüfung des definierten TLS-Namens auch Kanalauthentifizierungsdatensätze verwendet werden. Über die Authentifizierungsdatensätze für Kanäle können verschiedene Muster für definierte Namen des Zertifikatinhabers in TLS auf denselben Kanal angewendet werden. Wenn sowohl **SSLPEER** im Kanal als auch ein Kanalauthentifizierungsdatensatz für denselben Kanal verwendet werden, muss das eingehende Zertifikat mit beiden Mustern übereinstimmen, damit eine Verbindung hergestellt werden kann.

Wenn der vom Peer empfangene DN nicht mit dem Wert **SSLPEER** übereinstimmt, wird der Kanal nicht gestartet.

**SSLPEER** ist ein optionales Attribut. Ist hier kein Wert angegeben, wird der definierte Name des Peers beim Starten des Kanals nicht überprüft.

Die maximale Länge des Attributs **SSLPEER** hängt von der Plattform ab:

- **z/OS** Unter z/OS beträgt die maximale Länge des Attributs 256 Byte.
- **Multi** Auf allen anderen Plattformen beträgt sie 1024 Byte.

Kanalauthentifizierungsdatensätze bieten eine größere Flexibilität bei der Verwendung von **SSLPEER** und unterstützen eine maximale Länge von 1024 Byte auf allen Plattformen.

Die Überprüfung der **SSLPEER**-Attributwerte hängt auch von der Plattform ab:

- **z/OS** Unter z/OS werden die verwendeten Attributwerte nicht überprüft. Wenn Sie falsche Werte eingeben, schlägt der Kanal beim Starten fehl und an beiden Enden des Kanals werden in das Fehlerprotokoll Fehlermeldungen ausgegeben. Außerdem wird an beiden Enden des Kanals ein SSL-Fehler-Ereignis für den Kanal erstellt.
- **Multi** Auf anderen Plattformen als z/OS, die **SSLPEER** unterstützen, wird die Gültigkeit der Zeichenfolge bei der ersten Eingabe geprüft.

Sie können einen Wert für **SSLPEER** in einer Nicht-TLS-Kanaldefinition angeben, in der das Attribut **SSLCIPH** fehlt oder leer ist. In diesem Fall können Sie TLS temporär für die Fehlerbehebung inaktivieren, ohne die TLS-Parameter entfernen und später erneut eingeben zu müssen.

Weitere Informationen zur Verwendung von **SSLPEER** finden Sie unter [CHLAUTH FESTLEGEN](#) und [Schützen](#).

## Zugehörige Konzepte

[Kanalauthentifizierungsdatensätze](#)

## Themen-Root (TPROOT)

Dieses Attribut gibt den Themen-Root für einen AMQP-Kanal an.

Mit dem Attribut TPROOT können Sie einen Themen-Root für einen AMQP-Kanal angeben. Durch die Verwendung dieses Attributs ist sichergestellt, dass eine MQ Light-Anwendung bei ihrer Implementierung auf einem Warteschlangenmanager Nachrichten nicht in den Bereichen der Themenstruktur veröffentlicht, die von anderen Anwendungen verwendet werden, oder Nachrichten aus derartigen Bereichen subskribiert.

Der Standardwert für TPROOT ist SYSTEM.BASE.TOPIC. Bei diesem Wert weist die Themenzeichenfolge, die von einem AMQP-Client für die Veröffentlichung oder Subskription verwendet wird, kein Präfix auf und der Client kann Nachrichten mit anderen MQ-Publish/Subscribe-Anwendungen austauschen. Damit AMQP-Clients unter einem Themenpräfix Veröffentlichungen und Subskriptionen vornehmen können, müssen Sie zunächst ein MQ-Themenobjekt mit einer Themenzeichenfolge erstellen, die mit dem gewünschten Präfix festgelegt wird. Anschließend müssen Sie den Wert des Attributs TPROOT für den AMQP-Kanal in den Namen des von Ihnen erstellten MQ-Themenobjekts ändern. Im folgenden Beispiel ist ein Themen-Root dargestellt, der für den AMQP-Kanal MYAMQP auf den Wert APPGROUP1.BASE.TOPIC gesetzt wurde:

```
DEFINE CHANNEL(MYAMQP) CHLTYPE(AMQP) TPROOT(APPGROUP1.BASE.TOPIC) PORT(5673)
```

**Anmerkung:** Wenn der TPROOT-Attributwert oder die von ihm unterstützte Themenzeichenfolge geändert wird, kann dies dazu führen, dass bereits vorhandene AMQP-Themen und ihre Nachrichten verwaist sind.

## Name der Übertragungswarteschlange (XMITQ)

Dieses Attribut gibt den Namen der Übertragungswarteschlange an, aus der Nachrichten abgerufen werden.

Dieses Attribut ist für Kanäle vom Typ Sender und Server erforderlich und für andere Kanaltypen nicht gültig.

Geben Sie den Namen der Übertragungswarteschlange an, die diesem Sender- bzw. Serverkanal, der dem Warteschlangenmanager am fernen Ende des Kanals entspricht, zugeordnet werden muss. Sie können der Übertragungswarteschlange den gleichen Namen wie dem Warteschlangenmanager am fernen Ende geben.

Dieses Attribut ist für folgende Kanaltypen gültig:

- Sender
- Server

## Transporttyp (TRPTYPE)

Dieses Attribut gibt den zu verwendenden Transporttyp an.

Folgende Werte sind möglich:

LU62	LU 6.2
TCP	TCP/IP
NETBIOS	NetBIOS („1“ auf Seite 135)
SPX	SPX („1“ auf Seite 135)
<b>Anmerkungen:</b> 1. Zur Verwendung unter Windows. Kann auch unter z/OS zur Definition von Clientverbindungskanälen zur Verwendung unter Windows eingesetzt werden.	

Dieses Attribut gilt für alle Kanaltypen, wird jedoch von antwortenden Nachrichtenkanalagenten ignoriert.

## Client-ID verwenden (USECLTID)

Verwenden Sie die Client-ID für die Verbindung mit dem AMQP-Kanal.

Geben Sie an, ob die Client-ID für die Verbindung bei einem AMQP-Kanal verwendet werden soll. Setzen Sie dieses Attribut auf Yes (Ja) oder No (Nein).

## Warteschlange für nicht zustellbare Mail verwenden (USEDLQ)

Dieses Attribut bestimmt, ob die Warteschlange für nicht zustellbare Nachrichten (oder die Warteschlange für nicht zugestellte Nachrichten) verwendet wird, wenn Kanäle Nachrichten nicht übermitteln können.

Mögliche Werte:

### NO

Nachrichten, die von einem Kanal nicht zugestellt werden konnten, werden als Fehler behandelt. Entweder verwirft der Kanal die Nachricht oder der Kanal wird abgebrochen, entsprechend der NPMSPEED-Einstellung.

### YES (Standardwert)

Wenn das Attribut DEADQ des Warteschlangenmanagers den Namen einer Warteschlange für nicht zustellbare Nachrichten angibt, wird diese Warteschlange verwendet. Andernfalls ist das Verhalten wie bei NO.

### Benutzer-ID (USERID)

Dieses Attribut gibt die Benutzer-ID an, die der MCA beim Aufbauversuch einer sicheren SNA-Sitzung zu einem fernen MCA verwenden soll.

Sie können eine Task-Benutzer-ID mit 20 Zeichen angeben.

Es ist für die Kanaltypen Sender, Server, Requester und Clientverbindung gültig.

Dieses Attribut gilt mit Ausnahme von Clientverbindungskanälen nicht für IBM MQ for z/OS.

Wenn auf der Empfangsseite verschlüsselte Kennwörter verwendet werden und die LU 6.2-Software ein anderes Verschlüsselungsverfahren verwendet, schlägt der Kanalstart mit Einzelangaben zu ungültigen Sicherheitsbedingungen fehl. Dies kann vermieden werden, indem die empfangende SNA-Konfiguration dahingehend geändert wird, dass entweder

- die Kennwortersetzung inaktiviert wird oder
- eine Sicherheitsbenutzer-ID und das entsprechende Kennwort definiert werden.

Unter IBM MQ for z/OS ist dieses Attribut nur für Clientverbindungskanäle gültig. Auf anderen Plattformen ist es für folgende Kanaltypen gültig:


- Sender
- Server
- Requester
- Clientverbindung
- Clustersender

## IBM MQ-Clusterbefehle

---

Die zu den IBM MQ-Scriptbefehlen zählenden **runmqsc**-Befehle weisen spezielle Attribute und Parameter für Cluster auf. Es gibt aber auch andere Verwaltungsschnittstellen, die Sie zur Verwaltung von Clustern verwenden können.

Die MQSC-Befehle werden angezeigt, als ob Sie vom Systemadministrator in der Befehlskonsole eingegeben worden wären. Sie müssen diese Befehle aber nicht auf diese Weise ausgeben. Je nach Plattform gibt es verschiedene andere Methoden zur Ausgabe dieser Befehle, zum Beispiel:

- In IBM MQ for IBM i führen Sie MQSC-Befehle interaktiv über Option 26 von **WRKMQM** aus. Ebenso können Sie CL-Befehle verwenden oder MQSC-Befehle in einer Datei speichern und den CL-Befehl **STRMQMMQSC** verwenden.
-  Unter z/OS können Sie die Funktion COMMAND des Dienstprogramms **CSQUTIL**, die Bedienungs- und Steuerkonsolen oder die z/OS-Konsole verwenden.
- Auf allen anderen Plattformen können Sie die Befehle in einer Datei speichern und den Befehl **runmqsc** verwenden.

In einem MQSC-Befehl kann ein mit dem Attribut CLUSTER angegebener Clustername bis zu 48 Zeichen lang sein.

Eine mit dem Attribut CLUSNL angegebene Liste mit Clusternamen kann bis zu 256 Namen enthalten. Zur Erstellung einer Cluster-Namensliste verwenden Sie den Befehl **DEFINE NAMELIST**.



## IBM MQ Explorer

Die grafische Benutzerschnittstelle von IBM MQ Explorer kann einen Cluster mit Repository-Warteschlangenmanagern unter IBM WebSphere MQ for z/OS 6.0 oder höher verwalten. Die Benennung eines weiteren Repositorys auf einem separaten System ist nicht erforderlich. Bei früheren Versionen von IBM MQ for z/OS kann IBM MQ Explorer keinen Cluster mit Repository-Warteschlangenmanagern verwalten. Daher müssen Sie ein zusätzliches Repository auf einem System benennen, das von IBM MQ Explorer verwaltet werden kann.

Unter IBM MQ for Windows und IBM MQ for Linux können Sie IBM MQ Explorer auch zum Arbeiten mit Clustern einsetzen. Ebenso können Sie den eigenständigen IBM MQ Explorer-Client verwenden.

Bei Verwendung von IBM MQ Explorer können Sie Clusterwarteschlangen anzeigen und den Status von Clustersender- und Clusterempfängerkanälen abfragen. In IBM MQ Explorer gibt es zwei Assistenten, die Sie durch folgende Tasks führen:

- Cluster erstellen
- Unabhängigen Warteschlangenmanager mit einem Cluster verbinden

## Programmable Command Formats (PCF)

runmqsc-Befehl	PCF-Entsprechung
DISPLAY CLUSQMGR	MQCMD_INQUIRE_CLUSTER_Q_MGR
SUSPEND QMGR	MQCMD_SUSPEND_Q_MGR_CLUSTER
RESUME QMGR	MQCMD_RESUME_Q_MGR_CLUSTER
REFRESH CLUSTER	MQCMD_REFRESH_CLUSTER
RESET CLUSTER	MQCMD_RESET_CLUSTER

### Zugehörige Informationen

Clustering: Best Practices für REFRESH CLUSTER verwenden

## Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Um anzugeben, dass ein Warteschlangenmanager über ein vollständiges Repository für einen Cluster verfügt, verwenden Sie den Befehl ALTER QMGR unter Angabe des Attributs REPOS (*Clustername*). Um eine Liste mehrerer Clusternamen anzugeben, definieren Sie eine Clusternamensliste und verwenden Sie dann das Attribut REPOSNL (*Namensliste*) im Befehl ALTER QMGR:

```
DEFINE NAMELIST(CLUSTERLIST)
  DESCR('List of clusters whose repositories I host')
  NAMES(CLUS1, CLUS2, CLUS3)
ALTER QMGR REPOSNL(CLUSTERLIST)
```

Sie können im Befehl ALTER QMGR auch zusätzliche Clusterattribute angeben.

### CLWLEXIT (*Name*)

Gibt den Namen eines Benutzerexits an, der aufgerufen werden soll, wenn eine Nachricht in eine Clusterwarteschlange eingereicht wird.

### CLWLDATA (*Daten*)

Gibt die Daten an, die an den Benutzerexit für Clusterauslastung übergeben werden sollen.

### **CLWLLEN(Länge)**

Gibt die Höchstmenge an Nachrichtendaten an, die an den Benutzerexit für Clusterauslastung übergeben werden sollen.

### **CLWLMRUC(Kanäle)**

Gibt die maximale Anzahl an abgehenden Clusterkanälen an.

CLWLMRUC ist ein Attribut für einen lokalen Warteschlangenmanager, das nicht innerhalb des Clusters weitergegeben wird. Es wird den Exits für Clusterauslastung und dem Algorithmus für Clusterauslastung zur Verfügung gestellt, der die Zieladresse für Nachrichten auswählt.

### **CLWLUSEQ( LOCAL|ANY )**

Gibt das Verhalten der MQPUT-Operation an, wenn die Zielwarteschlange eine lokale Instanz und mindestens eine ferne Clusterinstanz besitzt. Geht der PUT-Vorgang von einem Clusterkanal aus, wird dieses Attribut nicht verwendet. Es ist möglich, CLWLUSEQ als Warteschlangenattribut und als Warteschlangenmanagerattribut anzugeben.

Wenn Sie die Option ANY angeben, sind sowohl die lokale Warteschlange als auch die fernen Warteschlangen mögliche Ziele der MQPUT-Operation.

Wenn Sie die Option LOCAL angeben, ist die lokale Warteschlange das einzige Ziel der MQPUT-Operation.

Die entsprechenden PCFs lauten MQCMD\_CHANGE\_Q\_MGR und MQCMD\_INQUIRE\_Q\_MGR.

## **Zugehörige Konzepte**

### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

## **Zugehörige Verweise**

### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

### DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl `RESET CLUSTER`, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## **Befehle für die Kanaldefinition**

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

Die Befehle DEFINE CHANNEL, ALTER CHANNEL und DISPLAY CHANNEL beinhalten zwei bestimmte CHLTYPE-Parameter für Cluster: CLUSRCVR und CLUSSDR. Um einen Clusterempfängerkanal zu definieren, verwenden Sie den Befehl DEFINE CHANNEL, wobei Sie CHLTYPE (CLUSRCVR) angeben. Viele der Attribute in einer Definition für einen Clusterempfängerkanal sind die gleichen Attribute wie die in einer Definition für einen Empfänger- oder Senderkanal. Um einen Clustersenderkanal zu definieren, verwenden Sie den Befehl DEFINE CHANNEL, wobei Sie CHLTYPE (CLUSSDR) angeben, sowie viele der Attribute, die Sie auch bei der Definition eines Senderkanals verwenden.

Für die Definition eines Clustersenderkanals ist es nicht mehr erforderlich, den Namen des vollständigen Warteschlangenmanager-Repositorys anzugeben. Wenn Ihnen die Namenskonvention bekannt ist, die für Kanäle in Ihrem Cluster verwendet wird, können Sie eine CLUSSDR-Definition mithilfe der Konstruktion +QMNAME+ vornehmen. Die +QMNAME+-Konstruktion wird unter z/OS nicht unterstützt. Nachdem die Verbindung hergestellt wurde, ändert IBM MQ den Namen des Kanals und ersetzt +QMNAME+ durch den korrekten Namen des vollständigen Warteschlangenmanager-Repositorys. Dieser Kanalname wird auf 20 Zeichen abgeschnitten.

Weitere Informationen zu Namenskonventionen finden Sie im Abschnitt [Namenskonventionen clustern](#).

Das Verfahren funktioniert nur dann, wenn Ihre Konvention für die Benennung von Kanälen den Namen des Warteschlangenmanagers miteinbezieht. Ein Beispiel: Sie definieren einen Warteschlangenmanager für ein vollständiges Repository mit der Bezeichnung QM1 in einem Cluster mit der Bezeichnung CLUSTER1 mit einem Clusterempfängerkanal CLUSTER1.QM1.ALPHA. Jeder andere Warteschlangenmanager kann einen Clustersenderkanal für diesen Warteschlangenmanager mit dem Kanalnamen CLUSTER1.QMNAME+.ALPHA definieren.

Wenn Sie für alle Ihre Kanäle die gleiche Namenskonvention verwenden, bedenken Sie, dass es immer nur eine +QMNAME+-Definition geben kann.

Die folgenden Attribute in den Befehlen DEFINE CHANNEL und ALTER CHANNEL gelten speziell für Clusterkanäle:

#### **Cluster**

Das Attribut CLUSTER gibt den Namen des Clusters an, dem dieser Kanal zugeordnet ist. Alternativ dazu können Sie das Attribut CLUSNL verwenden.

#### **CLUSNL**

Das Attribut CLUSNL gibt eine Namensliste mit Clusternamen an.

#### **NETPRTY**

Nur Clusterempfänger.

Das Attribut NETPRTY gibt eine Netzpriorität für den Kanal an. NETPRTY unterstützt die Auslastungsmanagementroutinen. Wenn es mehrere mögliche Routen zu einer Zieladresse gibt, wählt die Auslastungsmanagementroutine diejenige mit der höchsten Priorität aus.

#### **CLWLPRTY**

Der Parameter CLWLPRTY weist zu Auslastungsmanagementzwecken Kanälen mit der gleichen Zieladresse einen Prioritätsfaktor zu. Dieser Parameter gibt in Zusammenhang mit einer gleichmäßigen Clusterauslastung die jeweilige Priorität eines Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste Priorität und 9 die höchste Priorität bezeichnet.

#### **CLWLRANK**

Der Parameter CLWLRANK weist zu Auslastungsmanagementzwecken den Kanälen einen Rangfolgefaktor zu. Dieser Parameter gibt in Zusammenhang mit einer gleichmäßigen Clusterauslastung den jeweiligen Rang eines Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 den niedrigsten Rang und 9 den höchsten Rang bezeichnet.

#### **CLWLWGHT**

Der Parameter CLWLWGHT weist zum Zweck einer gleichmäßigen Clusterauslastung den Kanälen einen Gewichtungsfaktor zu. CLWLWGHT gewichtet den jeweiligen Kanal, sodass der Anteil der Nachrichten, die über diesen Kanal versendet werden, gesteuert werden kann. Der Algorithmus für Clusterenauslastung verwendet den Parameter CLWLWGHT, um die Auswahl der Zieladresse zu beeinflussen, sodass mehr Nachrichten über einen bestimmten Kanal versendet werden können. Alle Kanalgewichtungsattribute haben standardmäßig den gleichen Standardwert. Das Gewichtungsattribut ermöglicht es Ihnen, einem Kanal auf einer leistungsfähigen UNIX-Maschine eine stärkere Gewichtung zu geben

als einem Kanal auf einem kleinen Desktop-PC. Eine stärkere Gewichtung bedeutet, dass der Algorithmus für Clusterauslastung die UNIX-Maschine häufiger als Zieladresse für Nachrichten auswählt als den PC.

### **CONNNAME**

Der Parameter CONNNAME, der in der Definition für einen Clusterempfängerkanal angegeben ist, wird im gesamten Cluster verwendet, um die Netzadresse des Warteschlangenmanagers zu bestimmen. Achten Sie darauf, für den Parameter CONNNAME einen Wert auszuwählen, der in Ihrem gesamten IBM MQ-Cluster aufgelöst wird. Verwenden Sie keinen generischen Namen. Beachten Sie, dass der Wert, der im Clusterempfängerkanal angegeben wird, Vorrang vor jedem Wert hat, der in einem entsprechenden Clustersenderkanal angegeben wird.

Diese Attribute in den Befehlen DEFINE CHANNEL und ALTER CHANNEL gelten ebenso für den Befehl DISPLAY CHANNEL.

**Anmerkung:** Automatisch definierte Clustersenderkanäle erhalten ihre Attribute von der jeweiligen Definition für den entsprechenden Clusterempfängerkanal im Empfangswarteschlangenmanager. Auch wenn es sich um einen manuell definierten Clustersenderkanal handelt, werden seine Attribute automatisch geändert, um sicherzustellen, dass sie mit den Attributen der entsprechenden Clusterempfängerdefinition übereinstimmen. Sie können beispielsweise einen CLUSRCVR-Kanal definieren, ohne eine Portnummer im Parameter CONNNAME anzugeben, und dabei manuell einen CLUSSDR-Kanal definieren, bei dem eine Portnummer angegeben wird. Wenn der automatisch definierte CLUSSDR den manuell definierten Clustersenderkanal ersetzt, wird die Portnummer (vom CLUSRCVR übernommen) leer. In diesem Fall würde die Standardportnummer verwendet werden und der Kanal würde fehlschlagen.

**Anmerkung:** Der Befehl DISPLAY CHANNEL zeigt keine automatisch definierten Kanäle an. Sie können jedoch den Befehl DISPLAY CLUSQMGR verwenden, um die Attribute von automatisch definierten Clustersenderkanälen zu prüfen.

Verwenden Sie den Befehl DISPLAY CHSTATUS, um den Status von Clustersender- oder Clusterempfängerkanälen anzuzeigen. Dieser Befehl zeigt den Status sowohl von manuell definierten Kanälen als auch von automatisch definierten Kanälen an.

Die entsprechenden PCFs lauten MQCMD\_CHANGE\_CHANNEL, MQCMD\_COPY\_CHANNEL, MQCMD\_CREATE\_CHANNEL und MQCMD\_INQUIRE\_CHANNEL.

### **Weglassen des Werts CONNNAME in einer CLUSRCVR-Definition**

Unter Umständen können Sie den Wert CONNNAME in einer CLUSRCVR-Definition weglassen. Sie dürfen den Wert CONNNAME unter z/OS nicht übergehen.

**Multi** Unter Multiplatforms ist die Angabe des TCP/IP-Verbindungsnamensparameters eines Clusterempfängerkanals optional. Wenn kein Verbindungsname angegeben wird, generiert IBM MQ automatisch einen Verbindungsname, wobei der Standardport vorausgesetzt und die aktuelle IP-Adresse des Systems verwendet wird. Sie können die Standardportnummer überschreiben, aber die aktuelle IP-Adresse des Systems weiter verwenden. Lassen Sie für jeden Verbindungsname den IP-Namen leer und übergeben Sie die Portnummer in runden Klammern; Beispiel:

```
(1415)
```

Die generierte **CONNNAME** wird immer in der Schreibweise mit Trennzeichen (IPv4) oder im Hexadezimalformat (IPv6) und nicht in Form eines alphanumerischen DNS-Hostnamens generiert.

Diese Funktion ist hilfreich, wenn Ihre Systeme Dynamic Host Configuration Protocol (DHCP) verwenden. Wenn Sie für den Parameter CONNNAME in einem CLUSRCVR-Kanal keinen Wert angeben, müssen Sie die CLUSRCVR-Definition nicht ändern. DHCP ordnet Ihnen eine neue IP-Adresse zu.

Wenn Sie für den Parameter CONNNAME in der CLUSRCVR-Definition ein Leerzeichen angeben, generiert IBM MQ einen Wert für CONNNAME aus der IP-Adresse des Systems. Nur der generierte CONNNAME wird in den Repositories gespeichert. Andere Warteschlangenmanager im Cluster wissen nicht, dass für CONNNAME ursprünglich ein Leerzeichen angegeben war.

Wenn Sie den Befehl `DISPLAY CLUSQMGR` ausgeben, wird der generierte `CONNNAME` angezeigt. Wenn Sie jedoch den Befehl `DISPLAY CHANNEL` aus dem lokalen Warteschlangenmanager ausgeben, können Sie sehen, dass für `CONNNAME` ein Leerzeichen angegeben ist.

Wenn der Warteschlangenmanager aufgrund von DHCP gestoppt und mit einer anderen IP-Adresse neu gestartet wird, generiert IBM MQ `CONNNAME` erneut und aktualisiert die Repositorys entsprechend.

### Zugehörige Konzepte

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

#### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### Zugehörige Verweise

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

#### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

### Die Befehle **DEFINE QLOCAL**, **DEFINE QREMOTE** und **DEFINE QALIAS**

Die Clusterattribute in den Befehlen `DEFINE QLOCAL`, `DEFINE QREMOTE` und `DEFINE QALIAS` und die drei äquivalenten `ALTER`-Befehle sind:

#### **Cluster**

Gibt den Namen des Clusters an, dem die Warteschlange zugeordnet ist.

#### **CLUSNL**

Gibt eine Namensliste mit Clusternamen an.

#### **DEFBIND**

Gibt die Bindung an, die verwendet werden soll, wenn `MQ00_BIND_AS_Q_DEF` im `MQOPEN`-Aufruf einer Anwendung angegeben wird. Folgende Optionen stehen für dieses Attribut zur Verfügung:

- Wird `DEFBIND(OPEN)` angegeben, wird die Warteschlangenkenung beim Öffnen der Warteschlange an eine bestimmte Instanz der Clusterwarteschlange gebunden. `DEFBIND(OPEN)` ist die Standardeinstellung für dieses Attribut.
- Wird `DEFBIND(NOTFIXED)` angegeben, so wird die Warteschlangenkenung nicht an eine Instanz der Clusterwarteschlange gebunden.
- Wird `DEFBIND(GROUP)` angegeben, kann eine Anwendung fordern, dass alle Nachrichten einer Nachrichtengruppe an dieselbe Zielinstanz übergeben werden.

Wenn mehrere Warteschlangen mit demselben Namen in einem Warteschlangenmanagercluster zugänglich gemacht werden, können Anwendungen auswählen, ob alle Nachrichten aus dieser Anwendung an eine einzelne Instanz (`MQOO_BIND_ON_OPEN`) gesendet werden sollen, damit der Lastausgleichsalgorithmus für jede Nachricht das am besten geeignete Ziel auswählen kann (`MQOO_BIND_NOT_FIXED`), oder ob eine Anwendung anfordern kann, dass eine "Gruppe" von Nachrichten vollständig derselben Zielinstanz zugeordnet wird (`MQOO_BIND_ON_GROUP`). Der Lastausgleich wird zwischen Nachrichtengruppen erneut ausgeführt (ohne dass ein `MQCLOSE` und ein `MQOPEN` für die Warteschlange erforderlich sind).

Wenn Sie in einer Warteschlangendefinition `DEFBIND` angeben, wird die Warteschlange mit einem der Attribute `MQBND_BIND_ON_OPEN`, `MQBND_BIND_NOT_FIXED` oder `MQBND_BIND_ON_GROUP` definiert. Bei der Verwendung von Gruppen mit Clustern muss entweder `MQBND_BIND_ON_OPEN` oder `MQBND_BIND_ON_GROUP` angegeben werden.

Es wird empfohlen, für das Attribut `DEFBIND` in allen Instanzen einer Clusterwarteschlange denselben Wert anzugeben. Da `MQOO_BIND_ON_GROUP` in IBM WebSphere MQ 7.1 ein neues Attribut ist, darf dieses Attribut nicht verwendet werden, wenn eine der Anwendungen, die diese Warteschlange öffnen, eine Verbindung mit Warteschlangenmanagern aus IBM WebSphere MQ 7.0.1 oder früheren Versionen herstellt.

#### **CLWLRANK**

Weist zu Auslastungsmanagementzwecken Warteschlangen einen Rangfolgefaktor zu. Der `CLWLRANK`-Parameter wird von Modellwarteschlangen nicht unterstützt. Der Algorithmus für Clusterauslastung wählt eine Zielwarteschlange mit dem höchsten Rang aus. Standardmäßig wird `CLWLRANK` für alle Warteschlangen auf null gesetzt.

Wenn es sich bei dem Zielort um einen Warteschlangenmanager in einem anderen Cluster handelt, können Sie den Rang jedes zwischengeschalteten Gateway-Warteschlangenmanagers am Schnittpunkt benachbarter Cluster festlegen. Nachdem unter den zwischengeschalteten Warteschlangenmanagern eine Rangfolge festgelegt wurde, wählt der Algorithmus für Clusterauslastung ordnungsgemäß jeweils den Zielwarteschlangenmanager aus, der sich näher am Zielort befindet.

Das Gleiche gilt für Aliaswarteschlangen. Die Auswahl nach Rangfolge erfolgt noch, bevor der Kanalstatus überprüft wird. Daher stehen auch nicht verfügbare Warteschlangenmanager für die Auswahl zur Verfügung. Dadurch wird es einer Nachricht ermöglicht, durch das Netz weitergeleitet zu werden, anstatt dass sie zwischen zwei möglichen Zieladressen wählen muss (wie es bei der Prioritätseinstellung der Fall wäre). Wenn nun ein Kanal nicht in dem Bereich gestartet wird, den sein Rang angibt, wird die Nachricht nicht an den nächsthöheren Rang weitergeleitet, sondern wartet, bis ein Kanal für diese Zieladresse verfügbar ist (d. h. die Nachricht wird in der Übertragungswarteschlange gespeichert).

#### **CLWLPRTY**

Weist zu Auslastungsmanagementzwecken Warteschlangen einen Prioritätsfaktor zu. Der Algorithmus für Clusterauslastung wählt eine Zielwarteschlange mit der höchsten Priorität aus. Standardmäßig wird für alle Warteschlangen die Priorität null angegeben.

Bei zwei möglichen Zielwarteschlangen können Sie dieses Attribut dazu verwenden, eine Zieladresse als Ausfallsicherung für die andere Zieladresse einzurichten. Die Prioritätsauswahl erfolgt nach der Überprüfung des Kanalstatus. Alle Nachrichten werden an die Warteschlange mit der höchsten Priorität gesendet, es sei denn, der Status des Kanals zu dieser Zieladresse ist nicht so günstig wie der Status anderer Kanäle zu anderen Zieladressen. Dies bedeutet, dass nur die am besten zugänglichen Zieladressen für die Auswahl zur Verfügung stehen. Dadurch erfolgt eine Priorisierung unter mehreren Zieladressen, die alle verfügbar sind.

## CLWLUSEQ

Gibt das Verhalten einer MQPUT-Operation für eine Warteschlange an. Dieser Parameter gibt das Verhalten einer MQPUT-Operation an, wenn die Zielwarteschlange über eine lokale Instanz und mindestens eine ferne Clusterinstanz verfügt (außer wenn der MQPUT-Aufruf von einem Clusterkanal stammt). Dieser Parameter ist nur für lokale Warteschlangen gültig.

Folgende Werte sind möglich: QMGR (das Verhalten entspricht dem, was im CLWLUSEQ-Parameter der Warteschlangenmanagerdefinition angegeben ist), ANY (der Warteschlangenmanager behandelt die lokale Warteschlange zum Zweck der Lastverteilung wie eine weitere Instanz der Clusterwarteschlange) und LOCAL (die lokale Warteschlange ist das einzige Ziel der MQPUT-Operation, wobei angegeben wird, dass für die lokale Warteschlange das Einreihen aktiviert ist). Das Verhalten der MQPUT-Operation ist vom Algorithmus für das Clusterauslastungsmanagement abhängig.

## Befehle DISPLAY QUEUE und DISPLAY QCLUSTER

Die Attribute in den Befehlen DEFINE QLOCAL, DEFINE QREMOTE und DEFINE QALIAS gelten ebenso für den DISPLAY QUEUE-Befehl.

Um Informationen zu Clusterwarteschlangen anzuzeigen, geben Sie im Befehl DISPLAY QUEUE einen Warteschlangentyp wie QCLUSTER oder das Schlüsselwort CLUSINFO an, oder verwenden Sie den Befehl DISPLAY QCLUSTER.

Die Befehle DISPLAY QUEUE und DISPLAY QCLUSTER geben den Namen des Warteschlangenmanagers zurück, der die Warteschlange enthält (oder die Namen aller Warteschlangenmanager, wenn es sich um mehrere Instanzen einer Warteschlange handelt). Außerdem werden der Systemname für jeden Warteschlangenmanager, der die Warteschlange enthält, der dargestellte Warteschlangentyp sowie Datum und Uhrzeit des Zeitpunkts zurückgegeben, zu dem die Definition für den lokalen Warteschlangenmanager verfügbar gemacht wurde. Diese Daten werden mithilfe der Attribute CLUSQMGR, QMID, CLUSQT, CLUSDATE und CLUSTIME zurückgegeben.

Der Systemname für den Warteschlangenmanager (QMID) ist ein eindeutiger, systemgenerierter Name für den Warteschlangenmanager.

Sie haben die Möglichkeit, eine Clusterwarteschlange zu definieren, die gleichzeitig eine gemeinsam genutzte Warteschlange ist. Beispiel: Unter z/OS können Sie Folgendes definieren:

```
DEFINE QLOCAL(MYQUEUE) CLUSTER(MYCLUSTER) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

Die entsprechenden PCFs sind MQCMD\_CHANGE\_Q, MQCMD\_COPY\_Q, MQCMD\_CREATE\_Q und MQCMD\_INQUIRE\_Q.

### Zugehörige Konzepte

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

#### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### Zugehörige Verweise

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

#### DISPLAY CLUSQMGR



Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle `SUSPEND QMGR` und `RESUME QMGR` im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

#### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **`RESET CLUSTER`**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## **DISPLAY CLUSQMGR**

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

Wenn Sie diesen Befehl in einem Warteschlangenmanager mit einem vollständigen Repository ausgeben, gelten die zurückgegebenen Informationen für jeden Warteschlangenmanager im Cluster. Andernfalls gelten die zurückgegebenen Informationen nur für die jeweils betroffenen Warteschlangenmanager, d. h. für jeden Warteschlangenmanager, an den versucht wurde, eine Nachricht zu senden, und für jeden Warteschlangenmanager mit einem vollständigen Repository.

Die Daten beinhalten die meisten Kanalattribute sowohl für Clustersender- als auch für Clusterempfängerkanäle. Zusätzlich können die folgenden Attribute angezeigt werden:

#### **CHANNEL**

Der Name des Clusterempfängerkanals für den Warteschlangenmanager.

#### **CLUDDATE**

Das Datum, an dem die Definition dem lokalen Warteschlangenmanager zur Verfügung gestellt wurde.

#### **Cluster**

Gibt an, zu welchen Clustern der Warteschlangenmanager gehört.

#### **CLUSTIME**

Zeit, zu der die Definition für den lokalen Warteschlangenmanager verfügbar war.

#### **DEFTYPE**

Gibt die Definition des Warteschlangenmanagers an. Für `DEFTYPE` sind folgende Werte möglich:

#### **CLUSSDR**

Ein Clustersenderkanal wurde auf dem lokalen Warteschlangenmanager administrativ definiert, aber noch nicht vom Ziel-Warteschlangenmanager erkannt. Dieser Status tritt auf, wenn ein lokaler Warteschlangenmanager einen manuellen Clustersenderkanal definiert hat, der empfangende Warteschlangenmanager die Clusterinformationen jedoch nicht akzeptiert hat. Ursache hierfür kann sein, dass der Kanal aufgrund fehlender Verfügbarkeit oder eines Fehlers in der Clustersenderkonfiguration nicht aufgebaut wurde, zum Beispiel aufgrund einer fehlenden Übereinstimmung für die Eigenschaft `CLUSTER` in der Sender- und Empfängerdefinition. Hierbei handelt es sich um einen vorübergehenden Zustand oder Fehlerstatus, der untersucht werden sollte.

#### **CLUSSDRA**

Dieser Wert gibt einen automatisch erkannten Cluster-Warteschlangenmanager an, lokal ist kein Clustersenderkanal definiert. Dies ist der `DEFTYPE`-Wert für Cluster-Warteschlangenmanager, für die der lokale Warteschlangenmanager nicht über eine lokale Konfiguration verfügt, aber über die er informiert ist. Beispiel:

- Wenn der lokale Warteschlangenmanager ein Warteschlangenmanager für ein vollständiges Repository ist, muss der `DEFTYPE`-Wert für alle Teilrepository-Warteschlangenmanager im Cluster verwendet werden.



- Wenn der lokale Warteschlangenmanager ein Teilrepository ist, kann dies der Host einer Clusterwarteschlange sein, der von diesem lokalen Warteschlangenmanager oder von einem zweiten Warteschlangenmanager für ein vollständiges Repository verwendet wird, mit dem dieser Warteschlangenmanager zusammenarbeiten soll.

Wenn der DEFTYPE-Wert CLUSSDRA lautet und sowohl der lokale als auch der ferne Warteschlangenmanager vollständige Repositories für den benannten Cluster sind, ist die Konfiguration nicht korrekt, da ein lokal definierter Clustersenderkanal zum Umwandeln in den DEFTYPE-Wert CLUSSDRB definiert sein muss.

#### **CLUSSDRB**

Ein Clustersenderkanal wurde auf dem lokalen Warteschlangenmanager administrativ definiert und als gültiger Clusterkanal vom Ziel-Warteschlangenmanager erkannt. Dies ist der erwartete DEFTYPE-Wert für einen manuell konfigurierten Warteschlangenmanager für ein vollständiges Repository eines Teilrepository-Warteschlangenmanagers. Außerdem muss er der DEFTYPE-Wert für jeden CLUSQMGR von einem vollständigen Repository zu einem anderen vollständigen Repository im Cluster sein. Manuelle Clustersenderkanäle dürfen nicht zu Teilrepositories oder von einem Teilrepository-Warteschlangenmanager zu mehr als einem vollständigen Repository konfiguriert werden. Wenn der DEFTYPE-Wert CLUSSDRB in einer dieser Situationen festgestellt wird, muss er überprüft und korrigiert werden.

#### **CLUSRCVR**

Administrativ definiert als Clusterempfängerkanal für den lokalen Warteschlangenmanager. Gibt den lokalen Warteschlangenmanager im Cluster an.

**Anmerkung:** Informationen zum Ermitteln, welche CLUSQMGRs Warteschlangenmanager für vollständige Repositories für den Cluster sind, finden Sie unter der Eigenschaft QMTYPE.

Weitere Informationen zum Definieren von Clusterkanälen finden Sie unter Clusterkanäle.

#### **QMTYPE**

Gibt an, ob der Warteschlangenmanager über ein vollständiges Repository oder nur über ein Teilrepository verfügt.

#### **STATUS**

Gibt den Status des Clustersenderkanals für diesen Warteschlangenmanager an.

#### **SUSPEND**

Gibt an, ob der Warteschlangenmanager ausgesetzt wurde.

#### **Version**

Die Version der IBM MQ -Installation, der der Clusterwarteschlangenmanager zugeordnet ist.

Die Version hat das Format VVRRMMFF:

- VV: Version
- RR: Release
- MM: Wartungsstufe
- FF: Fix-Level

#### **XMITQ**

Die vom Warteschlangenmanager verwendete Clusterübertragungswarteschlange.

Siehe dazu den Befehl `DISPLAY QCLUSTER`. Er wird kurz in `DISPLAY QUEUE` und im Abschnitt Die Befehle DISPLAY QUEUE und DISPLAY QCLUSTER von „Befehle für die Warteschlangendefinition“ auf Seite 141 beschrieben. Beispiele zur Verwendung des Befehls `DISPLAY QCLUSTER` finden Sie in den Informationen zu `"DISPLAY QCLUSTER"` und `"DIS QCLUSTER"`.

#### **Zugehörige Konzepte**

##### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

## Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### **Zugehörige Verweise**

Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Use the SUSPEND QMGR and RESUME QMGR command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

REFRESH CLUSTER

Geben Sie den Befehl REFRESH CLUSTER in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

MQSC-Befehl **DISPLAY CLUSQMGR**

## **Befehle SUSPEND QMGR und RESUME QMGR im Cluster**

Use the SUSPEND QMGR and RESUME QMGR command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

Wenn ein Warteschlangenmanager auf einem Cluster ausgesetzt wird, empfängt er keine Nachrichten auf den gehosteten Clusterwarteschlangen, falls es eine verfügbare Warteschlange mit dem gleichen Namen auf einem alternativen Warteschlangenmanager im Cluster gibt. Werden Nachrichten jedoch explizit an diesen Warteschlangenmanager gerichtet oder ist die Zielwarteschlange nur auf diesem Warteschlangenmanager verfügbar, dann werden die Nachrichten weiterhin an diesen Warteschlangenmanager übertragen.

Der fortgesetzte Empfang eingehender Nachrichten im ausgesetzten Warteschlangenmanager kann verhindert werden, indem die Clusterempfängerkanäle für diesen Cluster gestoppt werden. Zum Stoppen der Clusterempfängerkanäle für einen Cluster verwenden Sie den Befehl SUSPEND QMGR im FORCE-Modus.

### **Zugehörige Konzepte**

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

## Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### **Zugehörige Tasks**

Verwalten eines Warteschlangenmanagers

### **Zugehörige Verweise**

Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

#### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### DISPLAY CLUSQMGR

Verwenden Sie den Befehl **DISPLAY CLUSQMGR**, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### REFRESH CLUSTER

Geben Sie den Befehl **REFRESH CLUSTER** in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

#### SUSPEND QMGR

#### RESUME QMGR

## REFRESH CLUSTER

Geben Sie den Befehl **REFRESH CLUSTER** in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

Es gibt drei Formen dieses Befehls:

#### **REFRESH CLUSTER(clustername) REPOS(NO)**

Die Standardeinstellung. Der Warteschlangenmanager hat Kenntnis von allen lokal definierten Clusterwarteschlangenmanagern und Clusterwarteschlangen sowie allen Clusterwarteschlangenmanagern, bei denen es sich um vollständige Repositories handelt. Wenn der Warteschlangenmanager darüber hinaus ein vollständiges Repository für den Cluster ist, hat er ebenso Kenntnis von den anderen Clusterwarteschlangenmanagern innerhalb des Clusters. Alles andere wird von der lokalen Kopie des Repositories gelöscht und aus den anderen vollständigen Repositories innerhalb des Clusters wiederhergestellt. Clusterkanäle werden nicht gestoppt, wenn **REPOS(NO)** verwendet wird. Ein vollständiges Repository verwendet seine **CLUSSDR**-Kanäle, um den Rest des Clusters darüber zu informieren, dass seine Aktualisierung abgeschlossen ist.

#### **REFRESH CLUSTER(clustername) REPOS(YES)**

Zusätzlich zum Standardverhalten werden Objekte, die Clusterwarteschlangenmanager für vollständige Repositories darstellen, ebenfalls aktualisiert. Diese Option kann nicht verwendet werden, wenn der Warteschlangenmanager ein vollständiges Repository ist; wird sie verwendet, schlägt der Befehl fehl und es wird der Fehler **AMQ9406/CSQX406E** protokolliert. Wenn es sich um ein vollständiges Repository handelt, müssen Sie es zuerst so ändern, dass es kein vollständiges Repository für den betreffenden Cluster ist. Die Adresse des vollständigen Repositories wird anhand der manuell definierten **CLUSSDR**-Definitionen wiederhergestellt. Nach Abschluss der Aktualisierung mit Angabe von **REPOS(YES)** kann der Warteschlangenmanager bei Bedarf geändert werden, sodass er wieder ein vollständiges Repository ist.

#### **REFRESH CLUSTER(\*)**

Aktualisiert den Warteschlangenmanager in allen Clustern, denen er angehört. In Verbindung mit **REPOS(YES)** zwingt **REFRESH CLUSTER(\*)** den Warteschlangenmanager dazu, die Suche nach vollständigen Repositories in den lokalen **CLUSSDR**-Definitionen erneut durchzuführen. Die Suche wird auch dann durchgeführt, wenn der **CLUSSDR**-Kanal den Warteschlangenmanager mit mehreren Clustern verbindet.

**Anmerkung:** Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle

interessierten Warteschlangenmanager hochladen, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken.

### Zugehörige Konzepte

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

#### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### Zugehörige Verweise

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

#### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

#### RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

### Zugehörige Informationen

Clustering: Best Practices für REFRESH CLUSTER verwenden

## RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **RESET CLUSTER**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

Sie können den Befehl **RESET CLUSTER** nur in Warteschlangenmanagern für ein vollständiges Repository ausgeben. Je nachdem, ob Sie mit Namen oder ID auf den Warteschlangenmanager verweisen, kann der Befehl zwei Formen annehmen.

1. 

```
RESET CLUSTER( clustername
) QMNAME( qmname ) ACTION(FORCEREMOVE) QUEUES(NO)
```

2. 

```
RESET CLUSTER( clustername
) QMID( qmid ) ACTION(FORCEREMOVE) QUEUES(NO)
```

Sie können nicht beide Optionen QMNAME und QMID angeben. Wenn Sie QMNAME verwenden, wird der Befehl nicht ausgeführt, wenn sich im Cluster mehrere Warteschlangenmanager mit diesem Namen befinden. Verwenden Sie QMID anstelle von QMNAME, um sicherzustellen, dass der Befehl **RESET CLUSTER** ausgeführt wird.

Bei der Angabe von QUEUES(NO) in einem **RESET CLUSTER**-Befehl handelt es sich um die Standardeinstellung. Durch die Angabe von QUEUES(YES) werden Verweise auf Clusterwarteschlangen entfernt, die zu den Warteschlangenmanagern des Clusters gehören. Diese Verweise werden zusätzlich zur Entfernung des Warteschlangenmanagers aus dem Cluster selbst entfernt.

Die Verweise werden entfernt, auch wenn der Clusterwarteschlangenmanager im Cluster nicht sichtbar ist, weil dessen Entfernung zuvor ohne die Option QUEUES erzwungen wurde.

Sie können den Befehl **RESET CLUSTER** verwenden, wenn beispielsweise ein Warteschlangenmanager gelöscht wurde, aber immer noch über für den Cluster definierte Clusterempfängerkanäle verfügt. Anstatt darauf zu warten, dass IBM MQ diese Definitionen entfernt (was automatisch geschieht), können Sie den Befehl **RESET CLUSTER** ausgeben, um die Definitionen bereits früher zu löschen. Alle anderen Warteschlangenmanager im Cluster werden daraufhin informiert, dass dieser Warteschlangenmanager nicht mehr verfügbar ist.

Wenn ein Warteschlangenmanager vorübergehend beschädigt ist, können Sie die anderen Warteschlangenmanager im Cluster benachrichtigen, bevor sie versuchen, Nachrichten an diesen Warteschlangenmanager zu senden. Der Befehl **RESET CLUSTER** entfernt den beschädigten Warteschlangenmanager. Später, wenn der beschädigte Warteschlangenmanager wieder funktioniert, verwenden Sie den Befehl **REFRESH CLUSTER**, um die Wirkung von **RESET CLUSTER** umzukehren und den Warteschlangenmanager an den Cluster zurückzugeben. Wenn sich der Warteschlangenmanager in einem Publish/Subscribe-Cluster befindet, müssen Sie alle erforderlichen Proxy-Subskriptionen wiederherstellen. Siehe [Hinweise zu REFRESH CLUSTER für Publish/Subscribe-Cluster](#).

**Anmerkung:** Bei großen Clustern kann der Befehl **REFRESH CLUSTER** während seiner Ausführung und danach in 27-Tage-Intervallen, wenn die Clusterobjekte ihre Statusaktualisierungen automatisch an alle interessierten Warteschlangenmanager hochladen, zu Unterbrechungen führen. Nähere Informationen hierzu erhalten Sie im Abschnitt [Die Aktualisierung in einem großen Cluster kann sich auf die Leistung und Verfügbarkeit auswirken](#).

Die Verwendung des Befehls **RESET CLUSTER** ist die einzige Möglichkeit, automatisch definierte Cluster-senderkanäle zu löschen.

**Wichtig:** Wenn der automatisch definierte Kanal, der entfernt werden soll, unbestätigt ist, entfernt **RESET CLUSTER** diesen Kanal nicht sofort. In dieser Situation müssen Sie vor dem Befehl **RESET CLUSTER** einen Befehl [RESOLVE CHANNEL](#) absetzen.

Es ist unwahrscheinlich, dass Sie diesen Befehl unter normalen Umständen benötigen. Das IBM Support Center rät Ihnen möglicherweise dazu, den Befehl abzusetzen, um die Clusterinformationen zu bereinigen, die in den Clusterwarteschlangenmanagern gespeichert sind. Verwenden Sie diesen Befehl nicht als verkürztes Verfahren, um Warteschlangenmanager aus einem Cluster zu entfernen. Die korrekte Vorgehensweise beim Entfernen eines Warteschlangenmanagers aus einem Cluster wird im Abschnitt [Warteschlangenmanager aus einem Cluster entfernen](#) beschrieben.

Repositorys speichern Daten nur für 90 Tage. Daher kann ein Warteschlangenmanager, dessen Entfernung erzwungen wurde, nach diesem Zeitraum die Verbindung zum Cluster wiederherstellen. Der Warteschlangenmanager stellt die Verbindung automatisch wieder her, es sei denn, er wurde gelöscht. Wenn Sie einen Warteschlangenmanager davon abhalten wollen, die Verbindung zu einem Cluster wiederherzustellen, müssen Sie entsprechende Sicherheitsmaßnahmen ergreifen.

Mit Ausnahme von **DISPLAY CLUSQMgr** agieren alle Clusterbefehle asynchron. Befehle zum Ändern von Objektattributen unter Einbeziehung von Clustering aktualisieren das Objekt und senden eine Anforderung an den Repositoryprozessor. Befehle für die Arbeit mit Clustern werden im Hinblick auf die Syntax überprüft und es wird eine Anforderung an den Repositoryprozessor gesendet.

Die an den Repositoryprozessor gesendeten Anforderungen werden gemeinsam mit den von anderen Clustermitgliedern empfangenen Clusteranforderungen asynchron verarbeitet. Die Verarbeitung nimmt

möglicherweise eine längere Zeit in Anspruch, wenn die Anforderungen an den gesamten Cluster gesendet werden müssen, um festzustellen, ob sie erfolgreich ausgeführt wurden oder nicht.

### **Zugehörige Konzepte**

#### Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

#### Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

### **Zugehörige Verweise**

#### Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

#### Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

#### Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

#### DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle SUSPEND QMGR und RESUME QMGR im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

#### REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### RESET CLUSTER (Cluster zurücksetzen)

## **Lastausgleich in Clustern**

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

Passende Zieladressen werden vom Verwaltungsalgorithmus für die Clusterauslastung auf Basis der Verfügbarkeit von Warteschlangenmanager und Warteschlange ausgewählt, sowie auf Basis einer Reihe spezifischer Attribute zur Clusterauslastung, die Warteschlangenmanagern, Warteschlangen und Kanälen zugeordnet sind. Diese Attribute werden in den Unterthemen beschrieben.

**Anmerkung:** Die Kanalattribute für die Clusterauslastung müssen Sie auf den Ziel-Warteschlangenmanagern auf den Clusterempfängerkanälen angeben. Lastausgleichsattribute, die Sie auf den zugehörigen Clustersenderkanälen angeben, werden vermutlich ignoriert. Siehe [Clusterkanäle](#).

Wenn die Konfiguration der spezifischen Attribute zur Clusterauslastung sich nicht wie erwartet verhält, untersuchen Sie die Einzelheiten zur Auswahl eines Warteschlangenmanagers durch den Algorithmus. Siehe „[Algorithmus für das Clusterauslastungsmanagement](#)“ auf Seite 155. Wenn die Ergebnisse dieses Algorithmus nicht Ihren Bedürfnissen entsprechen, können Sie ein Benutzerexitprogramm für Clusterauslastung schreiben und diesen Exit verwenden, um Nachrichten zu der Warteschlange Ihrer Wahl im Cluster weiterzuleiten. Weitere Informationen finden Sie unter [Exits für Clusterauslastung schreiben und kompilieren](#).



## Zugehörige Konzepte

 Asynchrones Verhalten von Clusterbefehlen unter z/OS

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

## Zugehörige Verweise

Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

Befehle `SUSPEND QMGR` und `RESUME QMGR` im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

REFRESH CLUSTER

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

RESET CLUSTER: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **`RESET CLUSTER`**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## Kanalattribute für Clusterauslastungsausgleich

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

### CLWLPRTY (Clusterauslastungspriorität)

Das Kanalattribut `CLWLPRTY` legt die Reihenfolge der Priorität der Kanäle für die Clusterlastverteilung fest. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Kanalattribut `CLWLPRTY` legen Sie die Reihenfolge der Priorität der verfügbaren Clusterziele fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Nachrichten gehen an den Warteschlangenmanager mit dem Kanal mit der höchsten Priorität. Ist dieser Kanal nicht verfügbar, gehen die Nachrichten an den Warteschlangenmanager mit der nächsthöchsten Priorität. Warteschlangenmanager mit niedrigerer Priorität fungieren als Reserve.

Vor der Priorisierung der Kanäle prüft IBM MQ den Kanalstatus. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

### Anmerkungen:

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt Clusterkanäle.
- Die Verfügbarkeit eines fernen Warteschlangenmanagers basiert auf dem Status des Kanals für diesen Warteschlangenmanager. Wenn Kanäle gestartet werden, ändert sich ihr Status mehrmals, wobei einige

Status für den Algorithmus für das Clusterauslastungsmanagement weniger günstig sind. In der Praxis bedeutet dies, dass Ziele mit einer niedrigeren Priorität (Sicherheit) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.

- Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherungsziel gesendet werden, verwenden Sie **CLWLPRTY** nicht. Ziehen Sie die Verwendung separater Warteschlangen in Betracht, oder **CLWLRANK** mit einem manuellen Umschalten von der primären auf die Sicherung.

## **CLWLRANK (Rangordnung der Clusterauslastung)**

Das Kanalattribut **CLWLRANK** gibt die Ebene der Kanäle für die Verteilung der Clusterauslastung an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Verwenden Sie das Kanalattribut **CLWLRANK**, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Sie steuern die Auswahl des Zielorts, indem Sie den Rang der Kanäle festlegen, die einen Warteschlangenmanager am Schnittpunkt der Cluster mit den Gateway-Warteschlangenmanagern verbinden.

Wenn **CLWLRANK** gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen. Ein Beispiel: Nachrichten kommen an einem Gateway-Warteschlangenmanager an, der sie an einen der beiden Warteschlangenmanager weiterleiten kann, die Kanäle mit dem Rang 1 und 2 verwenden. Die Nachrichten werden automatisch an den Warteschlangenmanager gesendet, der durch einen Kanal mit dem höchsten Rang verbunden ist; in diesem Fall ist dies der Warteschlangenmanagerkanal mit dem Rang 2.

IBM MQ ruft den Rang von Kanälen noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Kanäle zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Würden Sie hingegen auch das Prioritätsattribut **CLWLPRTY** verwenden, würde IBM MQ nur zwischen verfügbaren Zielen auswählen. Steht ein Kanal zum Ziel mit dem höchsten Rang nicht zur Verfügung, wird die Nachricht in der Übertragungswarteschlange zurückgehalten. Erst bei Verfügbarkeit des Kanals wird sie freigegeben. Die Nachricht wird also nicht an das nächste verfügbare Ziel der Rangordnung gesendet.

## **CLWLWGHT (Clusterauslastungsgewichtung)**

Das Kanalattribut **CLWLWGHT** gibt die Gewichtung von **CLUSDR**- und **CLUSRCVR**-Kanälen für eine gleichmäßige Clusterauslastung an. Der Wert muss zwischen 1 und 99 liegen, wobei 1 die niedrigste und 99 die höchste Gewichtung bezeichnet.

Verwenden Sie **CLWLWGHT**, um mehr Nachrichten an Server mit einer größeren Verarbeitungskapazität zu senden. Je stärker ein Kanal gewichtet ist, desto mehr Nachrichten werden über diesen Kanal versendet.

### **Anmerkungen:**

- Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).
- Die Standardeinstellung des Parameters **CLWLWGHT** ist 50. Wird diese Einstellung für einen Kanal geändert, so wird der Lastausgleich abhängig von der Häufigkeit, die die einzelnen Kanäle für eine an eine Clusterwarteschlange gesendete Nachricht ausgewählt wurden. Weitere Informationen finden Sie unter [„Algorithmus für das Clusterauslastungsmanagement“](#) auf Seite 155.



## NETPRTY (Netzverbindungspriorität)

Das Kanalattribut NETPRTY gibt die Priorität eines CLUSRCVR-Kanals an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Verwenden Sie das Attribut NETPRTY, um ein Netz als primäres Netz und ein weiteres Netz als Sicherungsnetz festzulegen. Bei einer Gruppe von Kanälen mit gleichem Rang wird durch Clustering der Pfad mit der höchsten Priorität ausgewählt, wenn mehrere Pfade zur Verfügung stehen.

Üblicherweise wird das Kanalattribut NETPRTY verwendet, um zwischen Netzen zu unterscheiden, die die gleichen Zielorte verbinden, aber unterschiedliche Aufwände verursachen oder unterschiedliche Geschwindigkeiten aufweisen.

**Anmerkung:** Geben Sie dieses Attribut auf dem Clusterempfängerkanal auf dem Ziel-Warteschlangenmanager an. Lastausgleichsattribute, die Sie auf dem zugehörigen Clustersenderkanal angeben, werden vermutlich ignoriert. Weitere Informationen finden Sie im Abschnitt [Clusterkanäle](#).

### Zugehörige Konzepte

[Algorithmus für das Clusterauslastungsmanagement](#)

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### Zugehörige Verweise

[Clusterauslastungsausgleich-Warteschlangenattribute](#)

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

[Clusterauslastungsausgleich-Warteschlangenmanager-Attribute](#)

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

## Clusterauslastungsausgleich-Warteschlangenattribute

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

## CLWLPRTY

Das Warteschlangenattribut **CLWLPRTY** gibt die Priorität einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 die niedrigste und 9 die höchste Priorität ist.

Mit dem Warteschlangenattribut **CLWLPRTY** legen Sie die Priorität einer Zielwarteschlange fest. IBM MQ wählt innerhalb des Clusters Zieladressen mit höherer Priorität vor Zieladressen mit niedrigerer Priorität aus. Falls mehrere Ziele die gleiche Priorität haben, wird das Ziel ausgewählt, das am längsten nicht verwendet wurde.

IBM MQ ruft die Priorität von Warteschlangenmanagern erst nach der Überprüfung des Kanalstatus ab. Nur verfügbare Warteschlangenmanager stehen zur Auswahl.

### Anmerkung:

Die Verfügbarkeit eines fernen Warteschlangenmanagers basiert auf dem Status des Kanals für diesen Warteschlangenmanager. Wenn Kanäle gestartet werden, ändert sich ihr Status mehrmals, wobei einige Status für den Algorithmus für das Clusterauslastungsmanagement weniger günstig sind. In der Praxis bedeutet dies, dass Ziele mit einer niedrigeren Priorität (Sicherung) ausgewählt werden können, während die Kanäle zu übergeordneten (primären) Zielen gestartet werden.

Wenn Sie sicherstellen müssen, dass keine Nachrichten an ein Sicherungsziel gesendet werden, verwenden Sie nicht **CLWLPRTY**. Ziehen Sie die Verwendung separater Warteschlangen in Betracht oder **CLWLRANK** mit einem manuellen Umschalten von der Primärdatenbank zur Sicherung.

Bei zwei möglichen Zielen können Sie dieses Attribut als Failover-Mechanismus verwenden. Anforderungen werden dem Warteschlangenmanager mit der höchsten Priorität zugestellt, während Warteschlan-

genmanager mit niedrigerer Priorität als Reserve fungieren. Fällt der Warteschlangenmanager mit der höchsten Priorität aus, übernimmt der Warteschlangenmanager mit der nächsthöheren Priorität, sofern er zur Verfügung steht.

## **CLWLRANK**

Das Warteschlangenattribut **CLWLRANK** gibt den Rang einer lokalen, fernen oder einer Aliaswarteschlange innerhalb der Arbeitslastverteilung in Clustern an. Der Wert muss zwischen 0 und 9 liegen, wobei 0 der niedrigste und 9 der höchste Rang ist.

Das Warteschlangenattribut **CLWLRANK** sollten Sie verwenden, wenn Sie die Zielorte von Nachrichten, die an einen Warteschlangenmanager in einem anderen Cluster gesendet werden, beeinflussen möchten. Wenn **CLWLRANK** gesetzt ist, werden Nachrichten über eine vorgegebene Route über die miteinander verbundenen Cluster an ein Ziel mit hohem Rang übertragen.

Stellen Sie sich zum Beispiel vor, Sie hätten zwei identisch konfigurierte Gateway-Warteschlangenmanager, durch die die Verfügbarkeit eines Gateways verbessert werden soll. Dazu hätten Sie an den Gateways Cluster-Aliaswarteschlangen für eine im Cluster definierte lokale Warteschlange konfiguriert. Sollte die lokale Warteschlange nicht zur Verfügung stehen, soll eine Nachricht an einem der Gateways zurückgehalten werden, bis die Warteschlange wieder zur Verfügung steht. Um die Cluster-Aliaswarteschlange an einem der beiden Gateways zurückzustellen, muss die lokale Warteschlange mit einem höheren Rang definiert sein als die Aliaswarteschlangen am Gateway.

Hätte die lokale Warteschlange den gleichen Rang wie die Aliaswarteschlangen und stünde die lokale Warteschlange nicht zur Verfügung, würde die Nachricht zwischen den Gateways hin und her übertragen werden. Sobald der erste Gateway-Warteschlangenmanager feststellt, dass die lokale Warteschlange nicht verfügbar ist, würde er die Nachricht an das andere Gateway senden. Das andere Gateway würde erneut versuchen, die Nachricht der lokalen Warteschlange zuzustellen. Stünde die lokale Warteschlange nach wie vor nicht zur Verfügung, würde die Nachricht zurück zum ersten Gateway geleitet werden. Die Nachricht wird auf diese Weise zwischen den beiden Gateways hin und her gesendet, bis die lokale Warteschlange wieder zur Verfügung steht. Erhält die lokale Warteschlange aber einen höheren Rang, würde die Nachricht selbst bei Nichtverfügbarkeit der Warteschlange keinem Ziel mit niedrigerem Rang zugestellt werden.

IBM MQ ruft den Warteschlangenrang noch vor der Überprüfung des Kanalstatus ab. Dies bedeutet, dass auch nicht verfügbare Warteschlangen zur Auswahl stehen. Dadurch können Nachrichten über das Netz weitergeleitet werden, selbst wenn das endgültige Ziel nicht zur Verfügung steht.

Wenn Sie hingegen das Prioritätsattribut 'CLWLPRTY' verwenden, wählt IBM MQ nur zwischen verfügbaren Zielen aus. Steht ein Kanal zum Ziel mit dem höchsten Rang nicht zur Verfügung, wird die Nachricht in der Übertragungswarteschlange zurückgehalten. Erst bei Verfügbarkeit des Kanals wird sie freigegeben. Die Nachricht wird also nicht an das nächste verfügbare Ziel der Rangordnung gesendet.

## **CLWLUSEQ**

Das Warteschlangenattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen als Zieladresse verwendet werden soll.

Das Warteschlangenattribut **CLWLUSEQ** ist nur für lokale Warteschlangen gültig. Es wird nur angewendet, wenn die Nachricht von einer Anwendung oder von einem Kanal eingereicht wird, der kein Clusterkanal ist.

## **LOCAL**

Das einzige Ziel von MQPUT ist die lokale Warteschlange, vorausgesetzt, diese ist PUT-aktiviert. Das Verhalten der MQPUT-Operation ist vom [Clusterauslastungsmanagement](#) abhängig.

## **QMGR**

Das Verhalten wird durch das Warteschlangenmanagerattribut **CLWLUSEQ** bestimmt.

## **Beliebig**

Hinsichtlich der Lastverteilung behandelt MQPUT die lokale Warteschlange genauso wie alle anderen Instanzen der Warteschlange im Cluster.

### **Zugehörige Konzepte**

Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### **Zugehörige Verweise**

Kanalattribute für Clusterauslastungsausgleich

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

Clusterauslastungsausgleich-Warteschlangenmanager-Attribute

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

## **Clusterauslastungsausgleich-Warteschlangenmanager-Attribute**

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

### **CLWLMRUC**

Das Warteschlangenmanagerattribut **CLWLMRUC** legt die Anzahl der zuletzt ausgewählten Kanäle fest. Der Verwaltungsalgorithmus für die Clusterauslastung verwendet **CLWLMRUC** zur Einschränkung der Anzahl der aktiven abgehenden Clusterkanäle. Der Wert muss im Bereich von 1 bis 999 999 999 liegen.

Der anfängliche Standardwert ist 999 999 999.

### **CLWLUSEQ**

Das Warteschlangenmanagerattribut **CLWLUSEQ** gibt an, ob innerhalb eines Clusters eine lokale Instanz einer Warteschlange vorrangig vor anderen Instanzen der Warteschlange als Ziel verwendet werden soll. Das Attribut wird angewendet, wenn das Warteschlangenattribut **CLWLUSEQ** auf QMGR gesetzt ist.

Das Warteschlangenattribut **CLWLUSEQ** ist nur für lokale Warteschlangen gültig. Es wird nur angewendet, wenn die Nachricht von einer Anwendung oder von einem Kanal eingereicht wird, der kein Clusterkanal ist.

### **LOCAL**

Das einzige Ziel von MQPUT ist die lokale Warteschlange. LOCAL ist der Standardwert.

### **Beliebig**

Hinsichtlich der Lastverteilung behandelt MQPUT die lokale Warteschlange genauso wie alle anderen Instanzen der Warteschlange im Cluster.

### **Zugehörige Konzepte**

Algorithmus für das Clusterauslastungsmanagement

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

### **Zugehörige Verweise**

Kanalattribute für Clusterauslastungsausgleich

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

Clusterauslastungsausgleich-Warteschlangenattribute

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

## **Algorithmus für das Clusterauslastungsmanagement**

Der Algorithmus für das Auslastungsmanagement wählt die Zielorte der in Clusterwarteschlangen befindlichen Nachrichten mittels Lastausgleichsattributen und Regeln aus.

Immer, wenn ein Ziel ausgewählt werden muss, wird der Workload-Management-Algorithmus ausgeführt:

- Sie wird an dem Punkt verwendet, an dem eine Clusterwarteschlange unter Verwendung der Option MQOO\_BIND\_ON\_OPEN geöffnet wird.
- Sie wird verwendet, wenn eine Nachricht in eine Clusterwarteschlange eingereicht wird, wenn sie mit MQOO\_BIND\_NOT\_FIXED geöffnet wird.
- Sie wird bei jedem Start einer neuen Nachrichtengruppe verwendet, wenn MQOO\_BIND\_ON\_GROUP verwendet wird, um eine Clusterwarteschlange zu öffnen.
- Er wird für Topic-Host-Routing bei jedem Veröffentlichen einer Nachricht für ein Clusterthema (Cluster-Topic) verwendet. Wenn es sich beim lokalen Warteschlangenmanager nicht um den Host für dieses Thema handelt, wird mit dem Algorithmus ein Host-Warteschlangenmanager ausgewählt, über den die Nachricht gesendet wird.

Im folgenden Abschnitt wird der Algorithmus für das Workload-Management beschrieben, der zur Bestimmung der Zielorte der in Clusterwarteschlangen befindlichen Nachrichten verwendet wird. Dessen Regeln werden durch die Einstellungen der folgenden Attribute für Warteschlangen, Warteschlangenmanager und Kanäle beeinflusst:

Tabelle 29. Attribute für das Clusterauslastungsmanagement		
Warteschlangen	Warteschlangenmanager	Kanäle
<ul style="list-style-type: none"> <li>• <a href="#">CLWLPRTY</a><sup>1</sup></li> <li>• <a href="#">CLWLRANK</a><sup>1</sup></li> <li>• <a href="#">CLWLUSEQ</a><sup>1</sup></li> <li>• PUT / PUB</li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">CLWLUSEQ</a><sup>1</sup></li> <li>• <a href="#">CLWLMRUC</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">CLWLPRTY</a></li> <li>• <a href="#">CLWLRANK</a></li> <li>• <a href="#">CLWLWGHT</a></li> <li>• <a href="#">NETPRTY</a></li> </ul>

Zunächst erstellt der Warteschlangenmanager mithilfe der folgenden beiden Verfahren eine Liste der möglichen Ziele:

- Abgleich des ObjectName und ObjectQmgrName des Ziels mit Warteschlangenmanager-Aliasdefinitionen, die in denselben Clustern wie der Warteschlangenmanager verwendet werden.
- Ermitteln eindeutiger Routen (also Kanäle) zu einem Warteschlangenmanager, der eine Warteschlange mit dem Namen ObjectName bereitstellt, die sich in einem der Cluster befindet, zu denen der Warteschlangenmanager gehört.

Der Algorithmus durchläuft die folgenden Regeln, um die Liste der möglichen Ziele zu reduzieren.

1. Ferne Instanzen von Warteschlangen oder Themen oder ferne CLUSRCVR-Kanäle, die keinen Cluster mit dem lokalen Warteschlangenmanager gemeinsam nutzen, werden ausgeschlossen.
2. Wenn eine Warteschlange oder ein Themenname angegeben ist, werden ferne CLUSRCVR-Kanäle ausgeschlossen, die sich nicht in demselben Cluster wie die Warteschlange oder das Thema befinden.

**Anmerkung:** Alle verbleibenden Warteschlangen, Themen und Kanäle werden in dieser Phase dem Exit für Clusterauslastung zur Verfügung gestellt, wenn dieser konfiguriert ist.

3. Alle Kanäle zu Warteschlangenmanagern oder Warteschlangenmanager-Aliasnamen, die einen CLWLRANK-Wert aufweisen, der kleiner ist als der maximale Rang aller verbleibenden Kanäle oder Warteschlangenmanager-Aliasnamen, werden ausgeschlossen.
4. Alle Warteschlangen (keine Warteschlangenmanager-Aliasnamen) mit einem CLWLRANK-Wert, der kleiner als der maximale Rang aller verbleibenden Warteschlangen ist, werden ausgeschlossen.
5. Wenn mehr als eine Instanz einer Warteschlange, eines Themas oder eines Warteschlangenmanager-Aliasnamens übrig bleibt und für mindestens eine davon 'pub put' aktiviert ist, werden alle Instanzen ausgeschlossen, auf denen 'put' inaktiviert ist.

<sup>1</sup> Dieses Attribut gilt bei Auswahl einer Clusterwarteschlange, nicht bei Auswahl eines Themas.

**Anmerkung:** Wenn ausschließlich Instanzen übrig bleiben, auf denen 'put' inaktiviert ist, können nur noch Abfrageoperationen erfolgreich ausgeführt werden, alle anderen Operationen schlagen mit MQRC\_CLUSTER\_PUT\_INHIBITED fehl.

6. Wenn bei der Auswahl einer Warteschlange die verbleibende Gruppe der Warteschlangen die lokale Instanz der Warteschlange enthält, wird in der Regel die lokale Instanz verwendet. Die lokale Instanz der Warteschlange wird verwendet, wenn eine der folgenden Bedingungen zutrifft:
  - Das Attribut CLWLUSEQ der Warteschlange ist auf LOCAL gesetzt.
  - Die beiden folgenden Aussagen treffen zu:
    - Das Attribut CLWLUSEQ der Warteschlange ist auf QMGR gesetzt.
    - Das Attribut CLWLUSEQ des Warteschlangenmanagers ist auf LOCAL gesetzt.
  - Die Nachricht wurde nicht direkt von einer lokalen Anwendung eingereicht, sondern über einen Clusterkanal empfangen.
  - Auf lokal definierte Warteschlangen, die mit CLWLUSEQ(ANY) definiert sind oder diese Einstellung aus dem Warteschlangenmanager übernehmen, treffen folgende Punkte innerhalb der breiter gesteckten Gruppe von geltenden Bedingungen zu:
    - Die lokale Warteschlange wird auf der Basis der status der lokal definierten CLUSRCVR-Kanäle in demselben Cluster wie die Warteschlange ausgewählt. Dieser Status wird mit dem Status der CLUSSDR-Kanäle verglichen, die die Nachricht an remote definierte gleichnamige Warteschlangen leiten würden.

Beispiel: Es gibt einen CLUSRCVR im gleichen Cluster wie die Warteschlange. Dieser CLUSRCVR hat den Status STOPPING, während die anderen Warteschlangen desselben Namens im Cluster den Status RUNNING oder INACTIVE haben. In diesem Fall werden die fernen Kanäle gewählt und die lokale Warteschlange wird nicht verwendet.
    - Die lokale Warteschlange wird auf der Basis der numbe von CLUSRCVR-Kanälen ausgewählt (im Vergleich zu CLUSSDR-Kanälen mit demselben Status), die die Nachricht an fern definierte Warteschlangen mit demselben Namen senden würden.

Beispiel: Es gibt vier CLUSRCVR-Kanäle im selben Cluster wie die Warteschlange und einen CLUSSDR-Kanal. Alle Kanäle haben denselben Status, d. h. entweder INACTIVE oder RUNNING. Daher stehen fünf Kanäle zur Auswahl und zwei Instanzen der Warteschlange. Vier Fünftel (80 Prozent) der Nachrichten gehen an die lokale Warteschlange.
7. Wenn mehr als ein Warteschlangenmanager übrig bleibt und mindestens einer davon nicht vorübergehend gesperrt ist, werden alle anderen ausgeschlossen, die vorübergehend gesperrt sind.
8. Wenn mehrere ferne Instanzen einer Warteschlange oder eines Themas verbleiben, werden alle Kanäle mit den Status INACTIVE oder RUNNING aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
  - MQCHS\_INACTIVE
  - MQCHS\_RUNNING
9. Wenn keine ferne Instanz einer Warteschlange oder eines Themas verbleibt, werden alle rev="d1">Kanäle mit den Status BINDING, INITIALIZING, STARTING oder STOPPING aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
  - MQCHS\_BINDING
  - MQCHS\_INITIALIZING
  - MQCHS\_STARTING
  - MQCHS\_STOPPING
10. Wenn keine Instanz einer Warteschlange oder eines Themas verbleibt, werden alle Kanäle eingeschlossen, für die ein erneuter Verbindungsversuch durchgeführt wird. Die Statuskonstante wird aufgeführt:
  - MQCHS\_RETRYING

11. Wenn keine ferne Instanz einer Warteschlange oder eines Themas verbleibt, werden alle Kanäle mit den Status REQUESTING, PAUSED oder STOPPED aufgelistet. Dazu werden folgende Statuskonstanten angezeigt:
  - MQCHS\_REQUESTING
  - MQCHS\_PAUSED
  - MQCHS\_STOPPED
  - MQCHS\_SWITCHING
12. Wenn mehr als eine ferne Instanz einer Warteschlange oder eines Themas auf einem beliebigen Warteschlangenmanager verbleibt, werden Kanäle mit dem höchsten NETPRTY-Wert für die einzelnen Warteschlangenmanager ausgewählt.
13. Alle verbleibenden Kanäle und Warteschlangenmanager-Aliasnamen mit Ausnahme der Kanäle und Aliasnamen mit der höchsten CLWLPRTY-Priorität werden ausgeschlossen. Verbleiben dabei Warteschlangenmanager-Aliasnamen, so bleiben die Kanäle zu diesem Warteschlangenmanager eingeschlossen.
14. Bei Auswahl einer Warteschlange:
  - Alle Warteschlangen mit Ausnahme der Warteschlangen mit der höchsten CLWLPRTY-Priorität werden ausgeschlossen, Kanäle bleiben aber erhalten.
15. Die verbleibenden Kanäle werden dann durch Entfernen der Kanäle mit den niedrigsten MQWDR.DestSeqNumber-Werten auf nicht mehr als die maximal zulässige Anzahl der zuletzt verwendeten Kanäle (CLWLMRUC) reduziert.

**Anmerkung:** Interne Steuernachrichten für das Cluster werden ggf. mit dem gleichen Algorithmus für die Clusterauslastung gesendet.

Nachdem die Liste der gültigen Ziele berechnet wurden, wird ein Lastausgleich für die Nachrichten mithilfe der folgenden Logik durchgeführt:

- Wenn mehrere ferne Instanzen eines Ziels verbleiben und für alle Kanäle zu diesem Ziel die Option CLWLWGHT auf die Standardeinstellung 50 gesetzt ist, wird der am längsten nicht mehr verwendete Kanal ausgewählt. Dies entspricht etwa einem Umlaufstil beim Durchführen des Lastausgleichs, wenn mehrere ferne Instanzen vorhanden sind.
- Wenn mehrere Instanzen eines Ziels verbleiben und für mindestens einen Kanal zu diesen Warteschlangen die Option CLWLWGHT auf eine andere Einstellung als die Standardeinstellung gesetzt ist (selbst wenn alle Kanäle einen übereinstimmenden, nicht standardmäßigen Wert haben), ist die Weiterleitung abhängig von den relativen Gewichtungen jedes Kanals und der Gesamtzahl, mit der jeder Kanal zuvor beim Senden von Nachrichten ausgewählt wurde.
- Beim Beobachten der Verteilung von Nachrichten für eine einzelne Clusterwarteschlange mit mehreren Instanzen führt dies scheinbar zu einer nicht ausgeglichenen Verteilung in einer Untergruppe der Warteschlangeninstanzen. Dies liegt daran, dass nicht nur die Nachrichtenübertragung für diese Warteschlange ausgeglichen wird, sondern die historische Verwendung jedes Clustersenderkanals aus diesem Warteschlangenmanager berücksichtigt wird. Wenn dieses Verhalten nicht erwünscht ist, führen Sie einen der folgenden Schritte aus:
  - Setzen Sie CLWLWGHT auf allen Clusterempfängerkanälen auf 50, wenn eine gleichmäßige Verteilung erforderlich ist.
  - oder, falls bestimmte Warteschlangeninstanzen anders als andere gewichtet werden müssen, definieren Sie diese Warteschlangen in einem dedizierten Cluster mit definierten dedizierten Clusterempfängerkanälen. Durch diese Aktion wird der Lastausgleich dieser Warteschlangen vom Lastausgleich anderer Warteschlangen im Cluster eingegrenzt.
- Die historischen Daten, mit denen der Lastausgleich für die Kanäle durchgeführt wird, werden zurückgesetzt, wenn ein beliebiges Attribut zur Clusterauslastung der verfügbaren Clusterempfängerkanäle geändert oder der Status eines Clusterempfängerkanals verfügbar wird. Durch Änderungen an den Auslastungsattributen manuell definierter Clustersenderkanäle werden die historischen Daten nicht zurückgesetzt.

- Wenn Sie erwägen, die Logik des Exits für die Clusterauslastung zu verwenden, wird der Kanal mit dem niedrigsten Wert für MQWDR.DestSeqFactor ausgewählt. Bei jeder Auswahl des Kanals wird dieser Wert um etwa 1000/CLWLWGHT erhöht. Falls mehrere Kanäle mit diesem Wert vorliegen, wird einer der Kanäle mit dem niedrigsten Wert für MQWDR.DestSeqNumber ausgewählt.

Die Verteilung der Benutzernachrichten erfolgt nicht immer exakt nach diesen Regeln, da auch durch die Clusterverwaltung Nachrichten über Kanäle übertragen werden. Dies führt zu einer zunächst ungleichmäßigen Verteilung der Benutzernachrichten, die sich erst nach einer gewissen Zeit stabilisiert. Aufgrund der Mischung aus Verwaltungs- und Benutzernachrichten sollten Sie sich nicht auf die genaue Verteilung von Nachrichten während des Lastausgleichs verlassen.

### **Zugehörige Verweise**

Kanalattribute für Clusterauslastungsausgleich

Eine alphabetische Liste der Kanalattribute, die beim Lastausgleich im Cluster verwendet werden.

Clusterauslastungsausgleich-Warteschlangenattribute

Eine alphabetische Liste der Warteschlangenattribute, die beim Lastausgleich im Cluster verwendet werden

Clusterauslastungsausgleich-Warteschlangenmanager-Attribute

Eine alphabetische Liste der Warteschlangenmanagerattribute, die beim Lastausgleich im Cluster verwendet werden

## **Asynchrones Verhalten von Clusterbefehlen unter z/OS**

Der Ausgeber eines Clusterbefehls unter z/OS erhält die Bestätigung, dass ein Befehl gesendet wurde, aber nicht darüber, ob er erfolgreich ausgeführt wurde.

Für beide Befehle REFRESH CLUSTER und RESET CLUSTER gilt, dass die Nachricht CSQM130I an den Ausgeber des Befehls gesendet wird, um anzuzeigen, dass eine Anforderung gesendet wurde. Darauf folgt die Nachricht CSQ9022I, um anzuzeigen, dass der Befehl erfolgreich ausgeführt wurde, d. h. dass eine Anforderung gesendet wurde. Sie gibt nicht an, dass die Clusteranforderung selbst erfolgreich ausgeführt wurde.

Alle eventuell auftretenden Fehler werden an die z/OS-Konsole des Systems ausgegeben, auf dem der Kanalinitiator aktiv ist; sie werden nicht an den Ausgeber des Befehls gemeldet.

Das asynchrone Verhalten steht im Kontrast zu Kanalbefehlen (CHANNEL). Sofort wird eine Nachricht ausgegeben, die anzeigt, dass ein Kanalbefehl akzeptiert wurde. Zu einem späteren Zeitpunkt wird, sobald der Befehl ausgeführt worden ist, eine Nachricht an den Ausgeber des Befehls gesendet, um eine normale oder abnormale Ausführung des Befehls anzuzeigen.

### **Zugehörige Konzepte**

Lastausgleich in Clustern

Enthält ein Cluster mehrere Instanzen der gleichen Warteschlange, wählt IBM MQ einen Warteschlangenmanager aus, an den eine Nachricht weitergeleitet werden soll. Dabei werden der Verwaltungsalgorithmus für die Clusterauslastung sowie eine Reihe spezifischer Attribute zur Clusterauslastung verwendet, um zu bestimmen, welcher Warteschlangenmanager sich am besten eignet.

### **Zugehörige Tasks**

Prüfen, ob asynchrone Befehle für verteilte Netze abgeschlossen sind

### **Zugehörige Verweise**

Befehle für Warteschlangenmanagerdefinitionen

Clusterattribute, die in den Befehlen für die Warteschlangenmanagerdefinition angegeben werden können.

Befehle für die Kanaldefinition

Clusterattribute, die in den Befehlen für die Kanaldefinition angegeben werden können.

Befehle für die Warteschlangendefinition

Clusterattribute, die in den Befehlen für die Warteschlangendefinition angegeben werden können.

DISPLAY CLUSQMGR

Verwenden Sie den Befehl `DISPLAY CLUSQMGR`, um Clusterinformationen zu Warteschlangenmanagern in einem Cluster anzuzeigen.

#### Befehle `SUSPEND QMGR` und `RESUME QMGR` im Cluster

Use the `SUSPEND QMGR` and `RESUME QMGR` command to temporarily reduce the inbound cluster activity to this queue manager, for example, before you perform maintenance on this queue manager, and then reinstate it.

#### `REFRESH CLUSTER`

Geben Sie den Befehl `REFRESH CLUSTER` in einem Warteschlangenmanager aus, um alle lokal gespeicherten Informationen zu einem Cluster zu löschen. Es ist unwahrscheinlich, dass Sie diesen Befehl unter anderen als außergewöhnlichen Umständen verwenden müssen.

#### `RESET CLUSTER`: Entfernen eines Warteschlangenmanagers aus einem Cluster erzwingen

Verwenden Sie den Befehl **`RESET CLUSTER`**, um in Ausnahmefällen einen Warteschlangenmanager zwangsweise aus einem Cluster zu entfernen.

## Kanalprogramme

In diesem Abschnitt werden die verschiedenen Arten von Kanalprogrammen (MCAs) behandelt, die bei den Kanälen verwendet werden können.

Die Namen der MCAs werden in der folgenden Tabelle aufgeführt.

*Tabelle 30. Kanalprogramme für Systeme mit Windows, UNIX and Linux*

Programmname	Verbindungsrichtung	Kommunikation
amqrmppa		Alle
runmqlsr	Eingehend	Alle
amqcrs6a	Eingehend	LU 6.2
amqcrsta	Eingehend	TCP
runmqchl	Ausgehend	Alle
runmqchi	Ausgehend	Alle

Die Steuerbefehle 'runmqlsr' (Ausführung des IBM MQ-Empfangsprogramms), 'runmqchl' (Ausführung des IBM MQ-Kanals) und 'runmqchi' (Ausführung des IBM MQ-Kanalinitiators) können in der Befehlszeile eingegeben werden.

'amqcrsta' wird für TCP-Kanäle auf Systemen unter UNIX and Linux mit 'inetd' aufgerufen, auf denen kein Empfangsprogramm gestartet wurde.

'amqcrs6a' wird bei Verwendung von LU6.2 als Transaktionsprogramm aufgerufen.

**IBM i**

## Jobs für übergreifende Kommunikation unter IBM i

Die folgenden Jobs sind der übergreifenden Kommunikation unter IBM i zugeordnet. Die Namen werden in der folgenden Tabelle aufgeführt.

*Tabelle 31. Jobnamen und -beschreibungen*

Jobname	Beschreibung
AMQCLMAA	Empfangsprogramm ohne Threads
AMQCRSTA	Responder-Job ohne Threads
AMQRMPPA	Channel-Pool-Job
RUNMQCHI	Kanalinitiator
RUNMQCHL	Channel-Job



Tabelle 31. Jobnamen und -beschreibungen (Forts.)

Jobname	Beschreibung
RUNMQLSR	Empfangsprogramm mit Threads

## IBM i Kanalzustände unter IBM i

Der Kanalstatus wird in der Anzeige "Work with Channels" (Mit Kanälen arbeiten) eingeblendet.

Tabelle 32. Kanalzustände unter IBM i

Statusname	Bedeutet
STARTING	Der Kanal ist zur Verhandlung mit dem Ziel-MCA bereit.
BINDING	Einrichten einer Sitzung und erster Datenaustausch
REQUESTING	Requesterkanal initialisiert eine Verbindung
RUNNING	Übertragung wird ausgeführt oder bereit für Übertragung
PAUSED	Warten auf Nachrichtenwiederholungs- intervall
STOPPING	Aufbau von Wiederholung oder Stopp
RETRYING	Warten auf nächsten Versuch
STOPPED	Kanal aufgrund eines Fehlers oder der Ausgabe eines "end-channel"-Befehls gestoppt
INACTIVE	Kanal hat die Verarbeitung normal beendet oder Kanal wurde nie gestartet
*Keine	Kein Status (nur für Serververbindungskanäle)

## ULW Beispiel für Nachrichtenkanalplanung für UNIX, Linux, and Windows

Dieser Abschnitt enthält ein ausführliches Beispiel der Herstellung einer Verbindung zwischen zwei Warteschlangenmanagern, damit sie untereinander Nachrichten austauschen können.

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM1 verwendet, Nachrichten in eine Warteschlange beim Warteschlangenmanager QM2 einreihen kann. Eine auf QM2 ausgeführte Anwendung kann diese Nachrichten abrufen und Antworten an eine Antwortwarteschlange auf QM1 senden.

Das Beispiel veranschaulicht die Verwendung von TCP/IP-Verbindungen. Dabei wird vorausgesetzt, dass Kanäle ausgelöst und gestartet werden sollen, sobald die erste Nachricht in der von ihnen bedienten Übertragungswarteschlange eingeht. Sie müssen den Kanalinitiator starten, damit der Auslösevorgang funktioniert.

In diesem Beispiel wird SYSTEM.CHANNEL.INITQ als Initialisierungswarteschlange verwendet. Diese Warteschlange ist in IBM MQ bereits definiert. Sie können zwar eine andere Initialisierungswarteschlange verwenden, müssen diese jedoch selbst definieren und beim Start des Kanalinitiators den Namen der Warteschlange angeben.

**Anmerkung:** V 9.1.4 MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf einen IBM Aspera fasp.io Gateway verweisen, der einen schnellen TCP/IP-Tunnel bereitstellt, mit dem der Netzdurchsatz erheblich gesteigert werden kann. Siehe [Aspera gateway -Verbindung unter Linuxdefinieren](#).

## ULW Erläuterung der Beispiele für UNIX, Linux, and Windows

Das Beispiel zeigt die IBM MQ-Befehle (MQSC), die Sie verwenden können.

In allen Beispielen werden die MQSC-Beispiele so angezeigt, wie sie in einer Befehlsdatei dargestellt und auch in einer Befehlszeile eingegeben würden. Die zwei Vorgehensweisen sehen identisch aus, aber um einen Befehl in die Befehlszeile einzugeben, müssen Sie zuerst `runmqsc` für den Standardwarteschlangenmanager bzw. `runmqsc qmname` eingeben, wobei `qmname` der Name der Name des betreffenden Warteschlangenmanagers ist. Geben Sie dann eine beliebige Anzahl Befehle ein, wie in den Beispielen gezeigt.

Eine alternative Möglichkeit besteht darin, eine Datei mit diesen Befehlen zu erstellen. Jegliche Fehler in den Befehlen können so leicht korrigiert werden. Wenn Sie Ihre Datei "mqsc.in" aufgerufen haben, um sie für den Warteschlangenmanager QMNAME auszuführen, geben Sie Folgendes ein:

```
runmqsc QMNAME < mqsc.in > mqsc.out
```

Durch folgende Eingabe können Sie die Befehle in Ihrer Datei verifizieren, bevor Sie sie ausführen:

```
runmqsc -v QMNAME < mqsc.in > mqsc.out
```

Die Zeilenlänge für Ihre Befehle sollte 72 Zeichen nicht übersteigen, um eine einwandfreie Portierbarkeit zu ermöglichen. Geben Sie ein Verkettungszeichen ein, wenn sich der Befehl über mehr als eine Zeile erstreckt. Verwenden Sie auf Windows-Systemen die Tastenkombination Strg + Z, um die Eingabe in die Befehlszeile zu beenden. Verwenden Sie auf Systemen mit UNIX and Linux die Tastenkombination Strg + D. Alternativ können Sie den Befehl **end** eingeben.

Abbildung 7 auf Seite 162 zeigt das Beispielszenario.

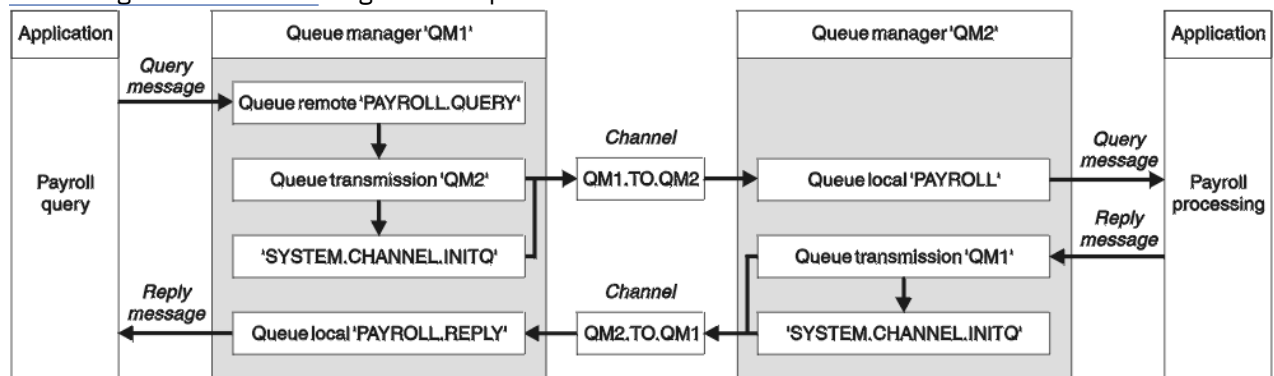


Abbildung 7. Nachrichtenkanalbeispiel für Systeme mit UNIX, Linux, and Windows

Im Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragen an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen. Die Lohnbuchhaltungsabfragen werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

In den Beispielfinitionen für TCP/IP hat QM1 die Hostadresse 192.0.2.0 und ist am Anschluss 1411 empfangsbereit, und QM2 hat die Hostadresse 192.0.2.1 und ist empfangsbereit am Anschluss 1412. Voraussetzung bei diesem Beispiel ist, dass diese Einstellungen bereits in Ihrem System vorgenommen wurden und die Warteschlangenmanager zur Verfügung stehen.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2
- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL
- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 7 auf Seite 162 zeigt ein entsprechendes Anordnungsdiagramm.

## Warteschlangenmanager QM1 - Beispiel für UNIX, Linux, and Windows

Über diese Objektdefinitionen können Anwendungen, die mit Warteschlangenmanager QM1 verbunden sind, Anforderungsnachrichten an die Warteschlange PAYROLL auf Warteschlangenmanager QM2 senden und Antworten in der Warteschlange PAYROLL.REPLY auf Warteschlangenmanager QM1 empfangen.

Alle Objektdefinitionen wurden mit den Attributen DESCR und REPLACE bereitgestellt. Die anderen genannten Attribute sind die Mindestvoraussetzung, die zum Ausführen des Beispiels erforderlich ist. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM1.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM1 aus.

### Definition der fernen Warteschlange

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +  
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQNAME(QM2)
```

**Anmerkung:** Die Definition einer fernen Warteschlange ist keine physische Warteschlange, sondern ein Mittel zum Übertragen von Nachrichten an die Übertragungswarteschlange QM2, damit sie an Warteschlangenmanager QM2 gesendet werden können.

### Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM1.TO.QM2.PROCESS)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den im namentlich genannten Prozess angegebenen Kanal.

## Senderkanaldefinition

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +  
CONNNAME('192.0.2.1(1412)')
```

## Empfängerkanaldefinition

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM2')
```

## Definition einer Empfangswarteschlange für Antworten

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to query messages sent to QM2')
```

Die Empfangswarteschlange für Antworten ist als PUT(ENABLED) definiert. Dies stellt sicher, dass Antwortnachrichten in die Warteschlange gestellt werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereiht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.

## Warteschlangenmanager QM2 - Beispiel für UNIX, Linux, and Windows

Die folgenden Objektdefinitionen ermöglichen Anwendungen mit Verbindung zum Warteschlangenmanager QM2, Anforderungsnachrichten aus der lokalen Warteschlange PAYROLL abzurufen und Antworten auf diese Anforderungsnachrichten in die Warteschlange mit dem PAYROLL.REPLY auf Warteschlangenmanager QM1 zu stellen.

Die Bereitstellung der Definition einer fernen Warteschlange zur Rückgabe der Antworten an QM1 entfällt hierbei. Der Nachrichtendeskriptor der aus der lokalen Warteschlange PAYROLL abgerufenen Nachricht enthält sowohl den Namen der Empfangswarteschlange für Antworten als auch den Namen des Managers der Empfangswarteschlange für Antworten. Solange also QM2 den Namen des Managers der Empfangswarteschlange für Antworten in den einer Übertragungswarteschlange auf Warteschlangenmanager QM2 auflösen kann, kann die Antwortnachricht gesendet werden. In diesem Beispiel lautet der Name des Managers der Empfangswarteschlange für Antworten QM1, daher benötigt Warteschlangenmanager QM2 eine Übertragungswarteschlange mit genau diesem Namen.

Alle Objektdefinitionen enthalten die Attribute DESCR und REPLACE, dies stellt eine Mindestvoraussetzung für ein Funktionieren des Beispiels dar. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM2.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM2 aus.

### Definition der lokalen Warteschlange

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Local queue for QM1 payroll details')
```

Diese Warteschlange ist aus demselben Grund wie die Definition der Empfangswarteschlange für Antworten auf Warteschlangenmanager QM1 als PUT(ENABLED) und GET(ENABLED) definiert.

## Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(QM2.TO.QM1.PROCESS)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den im namentlich genannten Prozess angegebenen Kanal.

### Senderkanaldefinition

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +
CONNAME('192.0.2.0(1411)')
```

### Empfängerkanaldefinition

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM1')
```

## ULW

## Beispiel für UNIX, Linux, and Windows ausführen

Informationen zum Starten des Kanalinitiators und des Empfangsprogramms sowie Vorschläge zum Erweitern dieses Szenarios.

Sobald diese Definitionen erstellt wurden, müssen Sie folgende Schritte ausführen:

- Starten Sie den Kanalinitiator für die einzelnen Warteschlangenmanager.
- Starten Sie das Empfangsprogramm für die einzelnen Warteschlangenmanager.

Informationen zum Starten des Kanalinitiators und des Empfangsprogramms finden Sie in den Abschnitten [Kommunikation für Windows konfigurieren](#) und [Kommunikation auf Systemen mit UNIX and Linux konfigurieren](#).

### Dieses Beispiel erweitern

Dieses einfache Beispiel lässt sich auf folgende Weise erweitern:

- Die Verwendung der LU 6.2-Kommunikation mit Verbindungen zwischen CICS-Systemen und Transaktionsverarbeitung.
- Hinzufügen weiterer Warteschlangen, Prozesse und Kanaldefinitionen, damit andere Anwendungen Nachrichten zwischen den zwei Warteschlangenmanagern übertragen können.
- Hinzufügen von Benutzerexitprogrammen auf Kanälen, um die Verbindungsverschlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden Sie Warteschlangenmanager-Aliasnamen und Aliasnamen für die Warteschlange für Antwortnachrichten, um weitere Kenntnisse darüber zu erlangen, wie diese in der Organisation Ihres Warteschlangenmanagernetzes verwendet werden können.

## IBM i

## Beispiel für Nachrichtenkanalplanung für IBM i

Dieser Abschnitt enthält ein ausführliches Beispiel für die Herstellung einer Verbindung zwischen zwei IBM i-Warteschlangenmanagern, damit sie untereinander Nachrichten austauschen können.

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM1 verwendet, Nachrichten in eine Warteschlange beim Warteschlangenmanager QM2 einreihen kann. Eine auf QM2 ausgeführte Anwendung kann diese Nachrichten abrufen und Antworten an eine Antwortwarteschlange auf QM1 senden.

Das Beispiel veranschaulicht die Verwendung von TCP/IP-Verbindungen. Dabei wird vorausgesetzt, dass Kanäle ausgelöst und gestartet werden sollen, sobald die erste Nachricht in der von ihnen bedienten Übertragungswarteschlange eingeht.

In diesem Beispiel wird SYSTEM.CHANNEL.INITQ als Initialisierungswarteschlange verwendet. Diese Warteschlange ist in IBM MQ bereits definiert. Sie können eine andere Initialisierungswarteschlange verwenden, müssen diese jedoch selbst definieren, mit dem Befehl STRMQMCHLI eine neue Instanz des Kanalinitiators starten und ihr den Namen Ihrer Initialisierungswarteschlange bereitstellen. Weitere Informationen zum Auslösen von Kanälen finden Sie im Abschnitt [Kanäle auslösen](#).

**Anmerkung:** V 9.1.4 MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf einen IBM Aspera fasp.io Gateway verweisen, der einen schnellen TCP/IP-Tunnel bereitstellt, mit dem der Netzdurchsatz erheblich gesteigert werden kann. Siehe [Aspera gateway -Verbindung unter Linuxdefinieren](#).

## IBM i Erläuterung der Beispiele für IBM i

In diesem Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragenachrichten an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen.

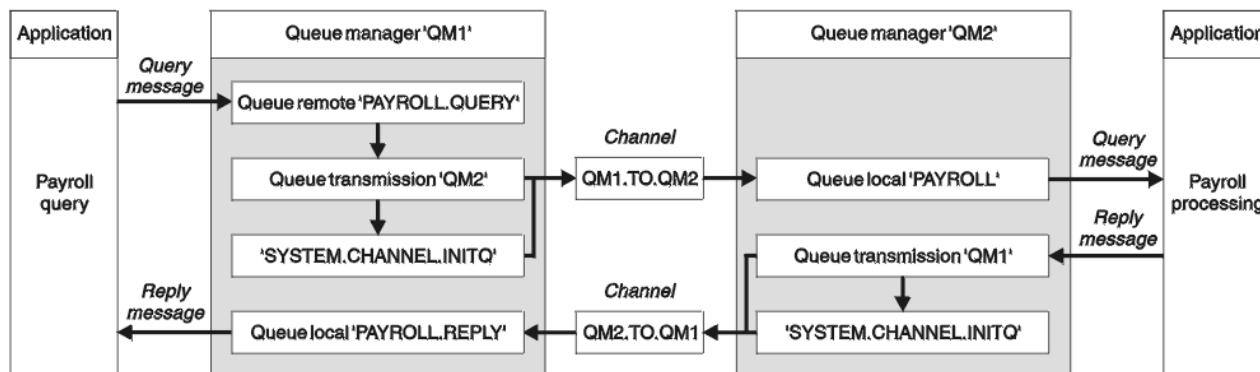


Abbildung 8. Nachrichtenkanalbeispiel für IBM MQ for IBM i

Die Lohnbuchhaltungsabfragenachrichten werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

Bei beiden Warteschlangenmanagern wird vorausgesetzt, dass sie unter IBM i ausgeführt werden. In den Beispielfinitionen hat QM1 die Hostadresse 192.0.2.0 und ist an Port 1411 empfangsbereit. QM2 hat die Hostadresse 192.0.2.1 und ist an Port 1412 empfangsbereit. In diesem Beispiel wird davon ausgegangen, dass diese Warteschlangenmanager bereits auf dem IBM i-System definiert wurden und die Warteschlangenmanager zur Verfügung stehen.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2
- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL
- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 8 auf Seite 166 zeigt ein entsprechendes Anordnungsdiagramm.

## Warteschlangenmanager QM1 - Beispiel für IBM i

Die folgenden Objektdefinitionen ermöglichen mit dem Warteschlangenmanager QM1 verbundenen Anwendungen, Anforderungsnachrichten an eine Warteschlange mit dem Namen PAYROLL auf QM2 zu senden, und Antworten in einer Warteschlange mit dem Namen PAYROLL.REPLY auf QM1 zu empfangen.

Sämtliche Objektdefinitionen werden in den TEXT-Attributen genannt. Die anderen genannten Attribute sind die Mindestvoraussetzung, die zum Ausführen des Beispiels erforderlich ist. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM1.

Führen Sie auf Warteschlangenmanager QM1 folgende Befehle aus:

### Definition der fernen Warteschlange

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	'PAYROLL.QUERY'
QTYPE	*RMT
TEXT	'Ferne Warteschlange für QM2'
PUTENBL	*YES
TMQNAME	'QM2' (Standard: Name des fernen Warteschlangenmanagers)
RMTQNAME	'PAYROLL'
RMTMQMNAME	'QM2'

**Anmerkung:** Die Definition einer fernen Warteschlange ist keine physische Warteschlange, sondern ein Mittel zum Übertragen von Nachrichten an die Übertragungswarteschlange QM2, damit sie an Warteschlangenmanager QM2 gesendet werden können.

### Definition der Übertragungswarteschlange

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	QM2
QTYPE	*LCL
TEXT	'Übertragungswarteschlange zu QM2'
USAGE	*TMQ

PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM1.TO.QM2

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den im namentlich genannten Prozess angegebenen Kanal.

#### **Senderkanaldefinition**

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM1.TO.QM2
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Senderkanal zu QM2'
TMQNAME	QM2
CONNNAME	'192.0.2.1(1412)'

#### **Empfängerkanaldefinition**

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM2.TO.QM1
CHLTYPE	*RCVR
TRPTYPE	*TCP
TEXT	'Empfängerkanal von QM2'

#### **Definition einer Empfangswarteschlange für Antworten**

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	PAYROLL.REPLY
QTYPE	*LCL
TEXT	'Antwortwarteschlange für Antworten zu Abfragenachrichten, die an QM2 gesendet wurden'
PUTENBL	*YES
GETENBL	*YES

Die Empfangswarteschlange für Antworten ist als PUT(ENABLED) definiert. Diese Definition stellt sicher, dass Antwortnachrichten in die Warteschlange eingereicht werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereicht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.



Die folgenden Objektdefinitionen ermöglichen Anwendungen mit Verbindung zum Warteschlangenmanager QM2, Anforderungsnachrichten aus der lokalen Warteschlange PAYROLL abzurufen und Antworten auf diese Anforderungsnachrichten in die Warteschlange mit dem PAYROLL.REPLY auf Warteschlangenmanager QM1 zu stellen.

Die Bereitstellung der Definition einer fernen Warteschlange zur Rückgabe der Antworten an QM1 entfällt hierbei. Der Nachrichtendeskriptor der aus der lokalen Warteschlange PAYROLL abgerufenen Nachricht enthält sowohl den Namen der Empfangswarteschlange für Antworten als auch den Namen des Managers der Empfangswarteschlange für Antworten. Solange also QM2 den Namen des Managers der Empfangswarteschlange für Antworten in den einer Übertragungswarteschlange auf Warteschlangenmanager QM2 auflösen kann, kann die Antwortnachricht gesendet werden. In diesem Beispiel lautet der Name des Managers der Empfangswarteschlange für Antworten QM1, daher benötigt Warteschlangenmanager QM2 eine Übertragungswarteschlange mit genau diesem Namen.

Alle Objektdefinitionen werden mit dem Attribut TEXT genannt und sind die Mindestvoraussetzung für die erfolgreiche Ausführung des Beispiels. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM2.

Führen Sie auf Warteschlangenmanager QM2 folgende Befehle aus:

#### **Definition der lokalen Warteschlange**

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	PAYROLL
QTYPE	*LCL
TEXT	'Lokale Warteschlange für Lohnbuchhaltungsdetails von QM1'
PUTENBL	*YES
GETENBL	*YES

Diese Warteschlange ist aus demselben Grund wie die Definition der Empfangswarteschlange für Antworten auf Warteschlangenmanager QM1 als PUT(ENABLED) und GET(ENABLED) definiert.

#### **Definition der Übertragungswarteschlange**

Der Befehl CRTMQMQ mit folgenden Attributen:

QNAME	QM1
QTYPE	*LCL
TEXT	'Übertragungswarteschlange zu QM1'
USAGE	*TMQ
PUTENBL	*YES
GETENBL	*YES
TRGENBL	*YES
TRGTYPE	*FIRST
INITQNAME	SYSTEM.CHANNEL.INITQ
TRIGDATA	QM2.TO.QM1

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den in den Auslöserdaten angegebenen Kanal.

#### **Senderkanaldefinition**

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM2.TO.QM1
CHLTYPE	*SDR
TRPTYPE	*TCP
TEXT	'Senderkanal zu QM1'
TMQNAME	QM1
CONNNAME	'192.0.2.0(1411)'

### Empfängerkanaldefinition

Der Befehl CRTMQMCHL mit folgenden Attributen:

CHLNAME	QM1.TO.QM2
CHLTYPE	*RCVR
TRPTYPE	*TCP
TEXT	'Empfängerkanal von QM1'

## IBM i Beispiel für IBM i ausführen

Nachdem Sie die erforderlichen Objekte erstellt haben, müssen Sie die Kanalinitiatoren und Empfangsprogramme für die beiden Warteschlangenmanager starten.

Die Anwendungen können sich daraufhin gegenseitig Nachrichten zusenden. Das Starten der Kanäle wird durch die erste Nachricht ausgelöst, die in jeder Übertragungswarteschlange eintrifft, daher muss der Befehl STRMQMCHL nicht ausgegeben werden.

Ausführliche Informationen zum Starten eines Kanalinitiators und eines Empfangsprogramms finden Sie im Abschnitt [Kanäle unter IBM i überwachen und steuern](#).

## IBM i Beispiel für IBM i erweitern

Das Beispiel kann auf zahlreiche Art erweitert werden.

Dieses Beispiel können Sie wie folgt erweitern:

- Fügen Sie weitere Warteschlangen- und Kanaldefinitionen hinzu, damit andere Anwendungen Nachrichten zwischen den beiden Warteschlangenmanagern senden können.
- Hinzufügen von Benutzerexitprogrammen auf Kanälen, um die Verbindungsverschlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden Sie Warteschlangenmanager-Aliasnamen und Aliasnamen für die Warteschlange für Antwortnachrichten, um weitere Kenntnisse darüber zu erlangen, wie diese Objekte in der Organisation Ihres Warteschlangenmanagernetzes verwendet werden können.

Eine Version dieses Beispiels, das MQSC-Befehle verwendet, finden Sie unter [„Beispiel für Nachrichtenkanalplanung für z/OS“](#) auf Seite 170.

## z/OS Beispiel für Nachrichtenkanalplanung für z/OS

Dieser Abschnitt enthält ein ausführliches Beispiel für die Herstellung einer Verbindung zwischen z/OS- oder MVS-Warteschlangenmanagern, damit sie untereinander Nachrichten austauschen können.

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM1 verwendet, Nachrichten in eine Warteschlange beim Warteschlangenmanager QM2 einreihen kann. Eine auf QM2 ausgeführte Anwendung kann diese Nachrichten abrufen und Antworten an eine Antwortwarteschlange auf QM1 senden.

Das Beispiel veranschaulicht die Verwendung von TCP/IP-Verbindungen und LU 6.2-Verbindungen. Dabei wird vorausgesetzt, dass Kanäle ausgelöst und gestartet werden sollen, sobald die erste Nachricht in der von ihnen bedienten Übertragungswarteschlange eingeht.

**Anmerkung:** V 9.1.4 MQ Adv. CD Ein Nachrichtenkanal, der TCP/IP verwendet, kann auf einen IBM Aspera fasp.io Gateway verweisen, der einen schnellen TCP/IP-Tunnel bereitstellt, mit dem der Netzdurchsatz erheblich gesteigert werden kann. Siehe [Aspera gateway -Verbindung unter Linuxdefinieren](#).

## z/OS Erläuterung der Beispiele für z/OS

In diesem Beispiel ist eine Anwendung zur Lohnbuchhaltungsabfrage mit Warteschlangenmanager QM1 verbunden, der Lohnbuchhaltungsabfragenachrichten an eine auf dem Warteschlangenmanager QM2 ausgeführte Anwendung zur Lohnbuchhaltungsabfrage sendet. Die Antworten auf die Abfragen der Anwendung zur Lohnbuchhaltungsabfrage müssen an QM1 erfolgen.

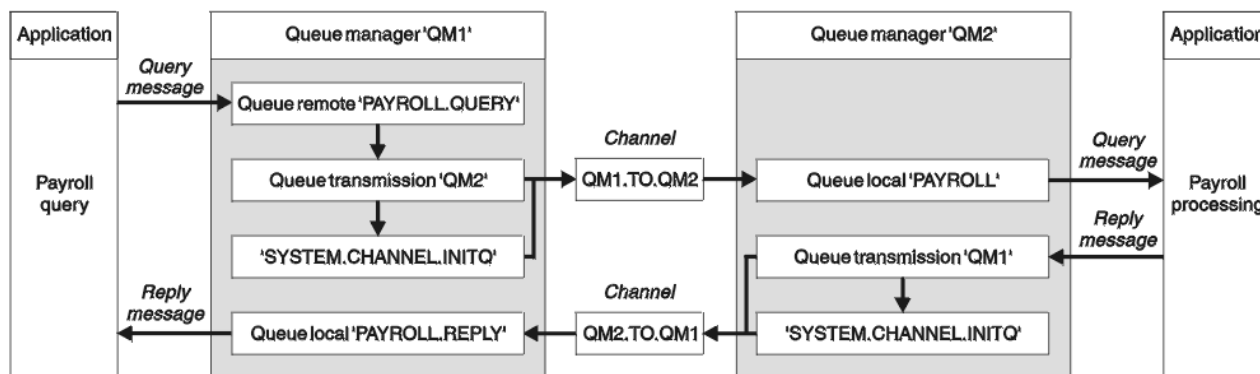


Abbildung 9. Erstes Beispiel für IBM MQ for z/OS

Die Lohnbuchhaltungsabfragenachrichten werden über den Sender-Empfänger-Kanal QM1.TO.QM2 von QM1 an QM2 gesendet, und die Antwortnachrichten werden auf einem weiteren Sender-Empfänger-Kanal mit dem Namen QM2.TO.QM1 von QM2 an QM1 gesendet. Das Starten beider Kanäle wird ausgelöst, sobald sie eine Nachricht erhalten, die an den anderen Warteschlangenmanager gesendet werden soll.

Die Anwendung zur Lohnbuchhaltungsabfrage reiht eine Abfragenachricht in die ferne Warteschlange "PAYROLL.QUERY" ein, die auf QM1 definiert ist. Diese Definition einer fernen Warteschlange wird in die lokale Warteschlange "PAYROLL" auf QM2 aufgelöst. Zusätzlich gibt die Anwendung zur Lohnbuchhaltungsabfrage an, dass die Antwort auf die Abfrage an die lokale Warteschlange "PAYROLL.REPLY" auf QM1 gesendet wird. Die Anwendung zur Lohnbuchhaltungsverarbeitung empfängt Nachrichten von der lokalen Warteschlange "PAYROLL" auf QM2 und sendet die Antworten dorthin, wo sie benötigt werden, in diesem Fall an die lokale Warteschlange "PAYROLL.REPLY" auf QM1.

Bei beiden Warteschlangenmanagern wird vorausgesetzt, dass sie unter z/OS ausgeführt werden. In den Beispielformatdefinitionen für TCP/IP hat QM1 die Hostadresse 192.0.2.0 und ist am Anschluss 1411 empfangsbereit, und QM2 hat die Hostadresse 192.0.2.1 und ist empfangsbereit am Anschluss 1412. In den Definitionen für LU 6.2 ist QM1 auf einer symbolischen logischen Einheit mit dem Namen LUNAME1 empfangsbereit und QM2 ist auf einer symbolischen logischen Einheit mit dem Namen LUNAME2 empfangsbereit. In diesem Beispiel wird davon ausgegangen, dass die Namen der logischen Einheiten (LUNAMES) bereits auf dem z/OS-System definiert wurden und die Warteschlangenmanager zur Verfügung stehen. Informationen zu ihrer Definition finden Sie in „IBM MQ-Beispielkonfiguration für z/OS“ auf Seite 49.

Folgende Objektdefinitionen müssen auf QM1 erstellt werden:

- Definition einer fernen Warteschlange, PAYROLL.QUERY
- Definition der Übertragungswarteschlange, QM2 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM1.TO.QM2

- Empfängerkanaldefinition, QM2.TO.QM1
- Definition der Empfangswarteschlange für Antworten, PAYROLL.REPLY

Folgende Objektdefinitionen müssen auf QM2 erstellt werden:

- Definition der lokalen Warteschlange, PAYROLL
- Definition der Übertragungswarteschlange, QM1 (Standardeinstellung = Name des fernen Warteschlangenmanagers)
- Senderkanaldefinition, QM2.TO.QM1
- Empfängerkanaldefinition, QM1.TO.QM2

In diesem Beispiel wird davon ausgegangen, dass alle SYSTEM.COMMAND.\* und SYSTEM.CHANNEL.\* Warteschlangen, die für die Ausführung von DQM erforderlich sind, wurden wie in den bereitgestellten Beispieldefinitionen **CSQ4INSG** und **CSQ4INSX** definiert.

Die Verbindungsdetails werden von dem Attribut CONNAME in den Senderkanaldefinitionen angegeben.

Abbildung 9 auf Seite 171 zeigt ein entsprechendes Anordnungsdiagramm.

## Warteschlangenmanager QM1 - Beispiel für z/OS

Die folgenden Objektdefinitionen ermöglichen mit dem Warteschlangenmanager QM1 verbundenen Anwendungen, Anforderungsnachrichten an eine Warteschlange mit dem Namen PAYROLL auf QM2 zu senden. Es ermöglicht Anwendungen ebenfalls, Antworten in einer Warteschlange auf QM1 mit dem Namen PAYROLL.REPLY zu empfangen.

Alle Objektdefinitionen wurden mit den Attributen DESCR und REPLACE bereitgestellt. Die anderen genannten Attribute sind die Mindestvoraussetzung, die zum Ausführen des Beispiels erforderlich ist. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM1.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM1 aus.

### Definition der fernen Warteschlange

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QM2') REPLACE +
PUT(ENABLED) XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

**Anmerkung:** Die Definition einer fernen Warteschlange ist keine physische Warteschlange, sondern ein Mittel zum Übertragen von Nachrichten an die Übertragungswarteschlange QM2, damit sie an Warteschlangenmanager QM2 gesendet werden können.

### Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM1.TO.QM2) INITQ(SYSTEM.CHANNEL.INITQ)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den in den Auslöserdaten angegebenen Kanal. Der Kanalinitiator kann nur Auslösenachrichten von der Warteschlange SYSTEM.CHANNEL.INITQ abrufen, verwenden Sie also keine andere Warteschlange als Initialisierungswarteschlange.

### Senderkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) +
```

```
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +  
CONNAME('192.0.2.1(1412)')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(LU62) +  
REPLACE DESCR('Sender channel to QM2') XMITQ(QM2) +  
CONNAME('LUNAME2')
```

## Empfängerkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM2')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(LU62) +  
REPLACE DESCR('Receiver channel from QM2')
```

## Definition einer Empfangswarteschlange für Antworten

```
DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Reply queue for replies to query messages sent to QM2')
```

Die Empfangswarteschlange für Antworten ist definiert als PUT(ENABLED), wodurch sichergestellt wird, dass Antwortnachrichten in die Warteschlange gestellt werden können. Wenn die Antworten nicht in die Empfangswarteschlange für Antworten eingereiht werden können, werden sie an die Warteschlange für nicht zustellbare Nachrichten auf QM1 gesendet bzw. bleiben, wenn diese Warteschlange nicht zur Verfügung steht, in der Übertragungswarteschlange QM1 auf Warteschlangenmanager QM2. Die Warteschlange wurde als GET(ENABLED) definiert, damit die Antwortnachrichten abgerufen werden können.

## Warteschlangenmanager QM2 - Beispiel für z/OS

Die folgenden Objektdefinitionen ermöglichen Anwendungen mit Verbindung zum Warteschlangenmanager QM2, Anforderungsnachrichten aus der lokalen Warteschlange PAYROLL abzurufen und Antworten auf diese Anforderungsnachrichten in die Warteschlange mit PAYROLL.REPLY auf Warteschlangenmanager QM1 zu stellen.

Die Bereitstellung der Definition einer fernen Warteschlange zur Rückgabe der Antworten an QM1 entfällt hierbei. Der Nachrichtendeskriptor der aus der lokalen Warteschlange PAYROLL abgerufenen Nachricht enthält sowohl den Namen der Empfangswarteschlange für Antworten als auch den Namen des Managers der Empfangswarteschlange für Antworten. Solange also QM2 den Namen des Managers der Empfangswarteschlange für Antworten in den einer Übertragungswarteschlange auf Warteschlangenmanager QM2 auflösen kann, kann die Antwortnachricht gesendet werden. In diesem Beispiel lautet der Name des Managers der Empfangswarteschlange für Antworten QM1, daher benötigt Warteschlangenmanager QM2 eine Übertragungswarteschlange mit genau diesem Namen.

Alle Objektdefinitionen enthalten die Attribute DESCR und REPLACE, dies stellt eine Mindestvoraussetzung für ein Funktionieren des Beispiels dar. Die Attribute, die nicht bereitgestellt werden, übernehmen die Standardwerte für Warteschlangenmanager QM2.

Führen Sie die folgenden Befehle für den Warteschlangenmanager QM2 aus.

## Definition der lokalen Warteschlange

```
DEFINE QLOCAL(PAYROLL) REPLACE PUT(ENABLED) GET(ENABLED) +  
DESCR('Local queue for QM1 payroll details')
```

Diese Warteschlange ist aus demselben Grund wie die Definition der Empfangswarteschlange für Antworten auf Warteschlangenmanager QM1 als PUT(ENABLED) und GET(ENABLED) definiert.

## Definition der Übertragungswarteschlange

```
DEFINE QLOCAL(QM1) DESCR('Transmission queue to QM1') REPLACE +  
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +  
TRIGDATA(QM2.TO.QM1) INITQ(SYSTEM.CHANNEL.INITQ)
```

Wird die erste Nachricht in diese Übertragungswarteschlange gestellt, wird eine Auslösenachricht an die Initialisierungswarteschlange, SYSTEM.CHANNEL.INITQ, gesendet. Der Kanalinitiator ruft die Nachricht aus der Initialisierungswarteschlange ab und startet den in den Auslöserdaten angegebenen Kanal. Der Kanalinitiator kann nur Auslösenachrichten aus SYSTEM.CHANNEL.INITQ abrufen, verwenden Sie also keine andere Warteschlange als Initialisierungswarteschlange.

## Senderkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(TCP) +  
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNAME('192.0.2.0(1411)')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM2.TO.QM1) CHLTYPE(SDR) TRPTYPE(LU62) +  
REPLACE DESCR('Sender channel to QM1') XMITQ(QM1) +  
CONNAME('LUNAME1')
```

## Empfängerkanaldefinition

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) +  
REPLACE DESCR('Receiver channel from QM1')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(LU62) +  
REPLACE DESCR('Receiver channel from QM1')
```

## Beispiel für z/OS ausführen

Wenn Sie die erforderlichen Objekte erstellt haben, müssen Sie die Kanalinitiatoren und die Empfangsprogramme für beide Warteschlangenmanager starten.

Die Anwendungen können sich daraufhin gegenseitig Nachrichten zusenden. Da der Start der Kanäle durch die Ankunft der ersten Nachricht in den einzelnen Übertragungswarteschlangen ausgelöst wird, müssen Sie den Befehl START CHANNEL MQSC nicht ausgeben.

Ausführliche Informationen zum Starten eines Kanalinitiators und eines Empfangsprogramms finden Sie in den Abschnitten [Kanalinitiator starten](#) bzw. [Kanalempfangsprogramm starten](#).

## Beispiel für z/OS erweitern

Das Beispiel kann auf zahlreiche Art erweitert werden.

Das Beispiel kann durch folgende Schritte erweitert werden:

- Hinzufügen weiterer Warteschlangen und Kanaldefinitionen, damit andere Anwendungen Nachrichten zwischen den zwei Warteschlangenmanagern übertragen können.
- Hinzufügen von Benutzerexitprogrammen auf Kanälen, um die Verbindungsverschlüsselung, die Sicherheitsprüfung und die Verarbeitung weiterer Nachrichten zu ermöglichen.
- Verwenden von Aliasen für Warteschlangenmanager und Aliasen für Empfangswarteschlangen für Antworten, um mehr darüber zu erfahren, wie diese Aliase in die Organisation Ihres Netzwerks aus Warteschlangenmanagern eingebunden werden können.

## Nachrichtenkanalplanungsbeispiel für z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange

Das Beispiel veranschaulicht die Vorbereitungen, die getroffen werden müssen, damit eine Anwendung, die den Warteschlangenmanager QM3 verwendet, Nachrichten in eine Warteschlange einer Gruppe mit gemeinsamer Warteschlange, die die Warteschlangenmitglieder QM4 und QM5 enthält, einreihen kann.

Bevor Sie dieses Beispiel ausprobieren, sollten Sie unbedingt mit dem Beispiel im Abschnitt [„Beispiel für Nachrichtenkanalplanung für z/OS“](#) auf Seite 170 vertraut sein.

## Erläuterung des Beispiels der Gruppe mit gemeinsamer Warteschlange für z/OS

Dieses Beispiel zeigt die IBM MQ-Befehle (MQSC), die Sie unter IBM MQ for z/OS in Zusammenhang mit Gruppen mit gemeinsamer Warteschlange für die verteilte Steuerung von Warteschlangen verwenden können.

Dieses Beispiel erweitert das Szenario "Lohnbuchhaltungsabfrage" des Beispiels in [„Beispiel für Nachrichtenkanalplanung für z/OS“](#) auf Seite 170 und zeigt, wie eine höhere Verfügbarkeit der Abfrageverarbeitung durch Hinzufügen weiterer bereitstellender Anwendungen hinzugefügt wird, um eine gemeinsam genutzte Warteschlange bereitzustellen.

Die Anwendung zur Lohnbuchhaltungsabfrage ist nun mit dem Warteschlangenmanager QM3 verbunden und reiht eine Abfrage in die in QM3 definierte ferne Warteschlange 'PAYROLL QUERY' ein. Diese definierte ferne Warteschlange wird in die gemeinsam genutzte Warteschlange PAYROLL aufgelöst, die sich auf den Warteschlangenmanagern in der Gruppe QSG1 mit gemeinsamer Warteschlange befindet. Die Anwendung zur Lohnbuchhaltungsverarbeitung hat nun zwei aktive Instanzen - eine ist mit QM4 verbunden, die andere mit QM5.

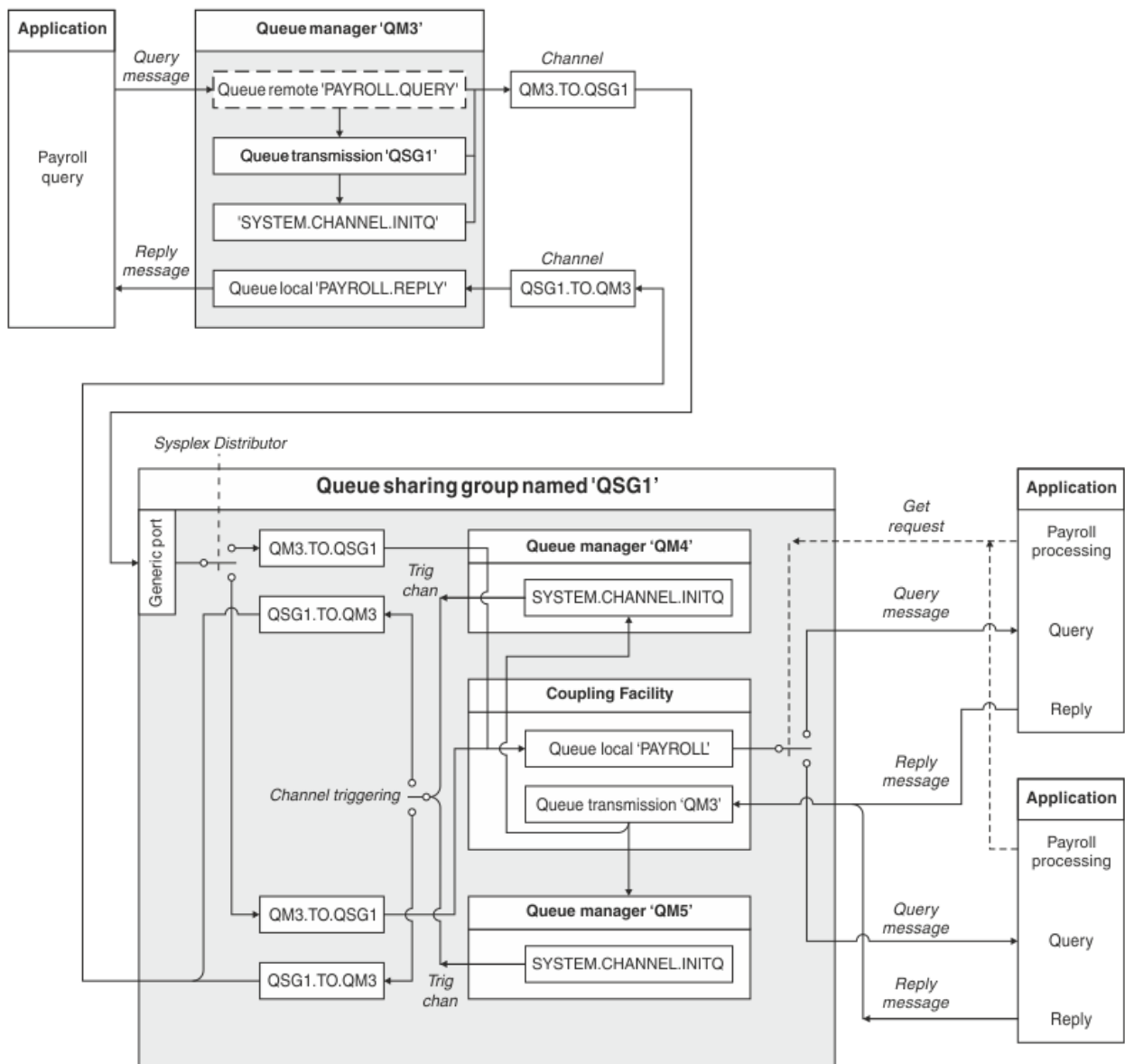


Abbildung 10. Nachrichtenkanalplanungsbeispiel für IBM MQ for z/OS unter Verwendung von Gruppen mit gemeinsamer Warteschlange

Bei allen drei Warteschlangenmanagern wird davon ausgegangen, dass sie unter z/OS aktiv sind. In den Beispielformen für TCP/IP hat QM4 die virtuelle IP-Adresse MVSIP01, QM5 die virtuelle IP-Adresse MVSIP02. Beide Warteschlangenmanager sind an Port 1414 empfangsbereit. Sysplex Distributor stellt für diese Gruppe die generische Adresse QSG1.MVSIP bereit. QM3 hat die Hostadresse 192.0.2.0 und ist an Port 1411 empfangsbereit.

In den Beispielformen für LU6.2 ist QM3 an dem symbolischen LU-Namen LUNAME1 empfangsbereit. Der Name der generischen Ressource, die für VTAM für die LU-Namen definiert ist, an denen QM4 und QM5 empfangsbereit sind, lautet LUQSG1. In diesem Beispiel wird davon ausgegangen, dass die Namen der logischen Einheiten (LUNAMES) bereits auf dem z/OS-System definiert wurden und verwendet werden können. Informationen zu ihrer Definition finden Sie unter „Selbstdefinition im Netz mithilfe generischer Ressourcen“ auf Seite 56.

In diesem Beispiel ist QSG1 der Name einer Gruppe mit gemeinsamer Warteschlange und die Namen der Warteschlangenmanager QM4 und QM5 sind Namen von Mitgliedern dieser Gruppe.



## Definitionen der Gruppe mit gemeinsamer Warteschlange

Durch Erstellen der folgenden Objektdefinitionen für ein Mitglied der Gruppe mit gemeinsamer Warteschlange stehen sie auch für alle anderen Mitglieder zur Verfügung.

Die Warteschlangenmanager QM4 und QM5 sind Mitglieder der Gruppe mit gemeinsamer Warteschlange. Die für QM4 erstellten Definitionen sind auch für QM5 verfügbar.

Es wird vorausgesetzt, dass die Listenstruktur der Coupling-Facility den Namen 'APPLICATION1' hat. Lautet der Name nicht 'APPLICATION1', verwenden Sie in dem Beispiel Ihren eigenen Namen der Listenstruktur der Coupling-Facility.

### Gemeinsam genutzte Objekte

Die Definitionen für gemeinsam genutzte Objekte sind in Db2 gespeichert, ihre zugehörigen Nachrichten sind in der Coupling-Facility gespeichert.

```
DEFINE QLOCAL(PAYROLL) QSGDISP(SHARED) REPLACE PUT(ENABLED) GET(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Shared queue for payroll details')

DEFINE QLOCAL(QM3) QSGDISP(SHARED) REPLACE USAGE(XMITQ) PUT(ENABLED) +
CFSTRUCT(APPLICATION1) +
DESCR('Transmission queue to QM3') TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QSG1.TO.QM3) GET(ENABLED) INITQ(SYSTEM.CHANNEL.INITQ)
```

### Gruppenobjekte

Die Gruppenobjektdefinitionen sind in Db2 gespeichert. Jeder Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange erstellt eine lokale Kopie des definierten Objekts.

Senderkanaldefinition für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNNAME('192.0.2.0(1411)')
```

Senderkanaldefinition für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(SDR) QSGDISP(GROUP) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QM3') XMITQ(QM3) +
CONNNAME('LUNAME1')
```

Empfängerkanaldefinition für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

Empfängerkanaldefinition für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QM3') QSGDISP(GROUP)
```

### Zugehörige Verweise

„Disposition (QSGDISP)“ auf Seite 113

Dieses Attribut gibt die Disposition des Kanals in einer Gruppe mit gemeinsamer Warteschlange an. Dieser Parameter ist nur unter z/OS gültig.

## Warteschlangenmanager QM3 - Beispiel für z/OS

QM3 ist kein Mitglied der Gruppe mit gemeinsamer Warteschlange. Die folgenden Objektdefinitionen ermöglichen ihm, Nachrichten in eine Warteschlange in der Gruppe mit gemeinsamer Warteschlange einzureihen.

Der CONNAME für diesen Kanal ist die generische Adresse der Gruppe mit gemeinsamer Warteschlange, die je nach Transporttyp variiert.

Für eine TCP/IP-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(TCP) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('QSG1.MVSIP(1414)')
```

Für eine LU 6.2-Verbindung:

```
DEFINE CHANNEL(QM3.TO.QSG1) CHLTYPE(SDR) TRPTYPE(LU62) +
REPLACE DESCR('Sender channel to QSG1') XMITQ(QSG1) +
CONNAME('LUQSG1') TPNAME('MQSERIES') MODENAME('#INTER')
```

### Sonstige Definitionen

Diese Definitionen sind für denselben Zweck wie die Definitionen im ersten Beispiel erforderlich.

```
DEFINE QREMOTE(PAYROLL.QUERY) DESCR('Remote queue for QSG1') REPLACE +
PUT(ENABLED) XMITQ(QSG1) RNAME(APPL) RQMNAME(QSG1)

DEFINE QLOCAL(QSG1) DESCR('Transmission queue to QSG1') REPLACE +
USAGE(XMITQ) PUT(ENABLED) GET(ENABLED) TRIGGER TRIGTYPE(FIRST) +
TRIGDATA(QM3.TO.QSG1) INITQ(SYSTEM.CHANNEL.INITQ)

DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(TCP) +
REPLACE DESCR('Receiver channel from QSG1')

DEFINE CHANNEL(QSG1.TO.QM3) CHLTYPE(RCVR) TRPTYPE(LU62) +
REPLACE DESCR('Receiver channel from QSG1')

DEFINE QLOCAL(PAYROLL.REPLY) REPLACE PUT(ENABLED) GET(ENABLED) +
DESCR('Reply queue for replies to query messages sent to QSG1')
```

## Beispiel mit Gruppe mit gemeinsamer Warteschlange für z/OS ausführen

Nachdem Sie die erforderlichen Objekte erstellt haben, müssen Sie die Kanalinitiatoren für alle drei Warteschlangenmanager starten. Gestartet werden müssen auch die Empfangsprogramme für beide Warteschlangenmanager in der Gruppe mit gemeinsamer Warteschlange.

Bei einer TCP/IP-Verbindung muss für jedes Mitglied der Gruppe ein Gruppenempfangsprogramm gestartet werden, das an Port 1414 empfangsbereit ist.

```
STA LSTR PORT(1414) IPADDR(MVSIP01) INDISP(GROUP)
```

Der vorherige Eintrag startet beispielsweise das Empfangsprogramm in QM4.

Bei einer LU 6.2-Verbindung muss für jedes Mitglied der Gruppe ein Gruppenempfangsprogramm gestartet werden, das an einem symbolischen LU-Namen empfangsbereit ist. Dieser LU-Name muss der generischen Ressource LUQSG1 entsprechen.

- Starten Sie das Empfangsprogramm in QM3.

```
STA LSTR PORT(1411)
```

## Einsatz eines Alias zum Verweis auf eine MQ-Bibliothek

Sie können ein Alias definieren, das auf eine MQ-Bibliothek in Ihrer Jobsteuersprache verweist, statt den Namen der MQ-Bibliothek direkt anzugeben. Wenn sich später der Name der MQ-Bibliothek ändert, brauchen Sie lediglich das Alias zu löschen und erneut zu definieren.

### Beispiel

Im folgenden Beispiel wird ein Alias MQM.SCSQANLE definiert, das auf die MQ-Bibliothek MQM.V600.SCSQANLE verweist:

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (MQM.SCSQANLE)
DEFINE ALIAS (NAME(MQM.SCSQANLE) RELATE(MQM.V600.SCSQANLE))
/*
```

Anschließend können Sie in Ihrer Jobsteuersprache über das Alias MQM.SCSQANLE auf die Bibliothek MQM.V600.SCSQANLE verweisen.

**Anmerkung:** Der Bibliotheksname und der Aliasname müssen sich in demselben Katalog befinden; verwenden Sie deshalb für beide dasselbe übergeordnete Qualifikationsmerkmal (im vorliegenden Beispiel MQM).

## Managed File Transfer - Konfigurationsreferenz

Die Referenzinformationen unterstützen Sie bei der Konfiguration von Managed File Transfer.

### Verwendung von Umgebungsvariablen in MFT-Eigenschaften

Ab IBM WebSphere MQ 7.5 können in Managed File Transfer-Eigenschaften Umgebungsvariablen für Datei- oder Verzeichnispositionen verwendet werden. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt).

In den folgenden Eigenschaften können Datei- oder Verzeichnispositionen auch in Form von Umgebungsvariablen angegeben werden:

- agentSslKeyStore
- agentSslKeyStoreCredentialsFile
- agentSslTrustStore
- agentSslTrustStoreCredentialsFile
- cdNodeKeystoreCredentialsFile
- cdNodeTruststoreCredentialsFile
- cdTmpDir
- cdNodeKeystore
- cdNodeTruststore
- commandPath
- connectionSslKeyStore
- connectionSslKeyStoreCredentialsFile
- connectionSslTrustStore

- `connectionSslTrustStoreCredentialsFile`
- `coordinationSslKeyStore`
- `coordinationSslKeyStoreCredentialsFile`
- `coordinationSslTrustStore`
- `coordinationSslTrustStoreCredentialsFile`
- `exitClassPath`
- `exitNativeLibraryPath`
- `javaCoreTriggerFile`
- `sandboxRoot`
- `transferRoot`
- `wmqfte.database.credentials.file`

### Beispiel für Windows

**Windows** Im nachfolgenden Beispiel für ein Windows-System verwendet der Benutzer `fteuser` die Umgebungsvariable `USERPROFILE`:

```
wmqfte.database.credentials.file=%USERPROFILE%\logger\mqmftcredentials.xml
```

Dies wird in folgenden Dateipfad aufgelöst:

```
C:\Users\fteuser\logger\mqmftcredentials.xml
```

### Beispiel für UNIX

**UNIX** Im nachfolgenden Beispiel für ein UNIX-System verwendet der Benutzer `fteuser` die Umgebungsvariable `HOME`:

```
transferRoot=$HOME/fte/
```

Dies wird in folgenden Dateipfad aufgelöst:

```
/home/fteuser/fte/
```

## Die MFT-Datei 'installation.properties'

Die Datei `installation.properties` gibt den Namen der Standardgruppe von Konfigurationsoptionen an. Dieser Eintrag verweist Managed File Transfer an eine strukturierte Gruppe mit Verzeichnissen und Eigenschaftendateien, welche die zu verwendende Konfiguration enthält. Gewöhnlich ist der Name einer Gruppe von Konfigurationsoptionen der Name des zugeordneten Koordinations-WS-Managers.

Diese Datei wird vom Installationsprogramm erstellt und kann über den Befehl **`fteChangeDefaultConfigurationOptions`** geändert werden.

Die Datei `installation.properties` befindet sich im Ihrem Verzeichnis `MQ_DATA_PATH`. Unter Windows ist die Standarddateiposition beispielsweise `MQ_DATA_PATH\mqft\installations\installation_name` und auf UNIX- und Linux-Systemen ist die Standarddateiposition `/var/mqm/mqft/installations/installation_name`.

**V 9.1.0** Für den Redistributable Managed File Transfer Agent wird der Datenpfad bei der Ausführung des Befehls **`fteCreateEnvironment`** gesetzt. Wird der Befehl unter Angabe des gewünschten Verzeichnisses (Parameter **`-d`**) ausgeführt, wird der Datenpfad auf diesen Wert gesetzt. Wenn Sie im Befehl **`fteCreateEnvironment`** keinen Datenpfad angeben, wird im Stammverzeichnis das Verzeichnis

mftdata erstellt, in dem der Redistributable Managed File Transfer Agent dann extrahiert wird. Die Datei `installation.properties` für den Redistributable Managed File Transfer Agent befindet sich im Verzeichnis `MQ_DATA_PATH\mqft\installations\MFTZipInstall`.

Die Datei `installation.properties` enthält die folgenden Werte:

Tabelle 33. Grundlegende Eigenschaften

Eigenschaftsname	Beschreibung	Standardwert
<p>commandMessagePriority</p>	<p>Legt die Priorität sowohl von internen Nachrichten als auch von Befehlsnachrichten für die Befehle <b>fteStopAgent</b>, <b>fteCancelTransfer</b> und <b>ftePingAgent</b> fest.</p> <p>Wenn Sie zum Beispiel sehr viele Übertragungsanforderungen zur Übertragung vieler kleiner Dateien in kurzer Folge übergeben, werden die neuen Übertragungsanforderungen möglicherweise in die Befehlswarteschlange des Quellagenten eingereiht. Für die externen und internen Nachrichten gilt die standardmäßige IBM MQ-Nachrichtenpriorität, sodass die internen Nachrichten durch die neuen Übertragungsanforderungen blockiert werden. Dies kann dazu führen, dass die zulässige Zeit für die Übertragungsvereinbarung überschritten wird und die Übertragungen in den Wiederherstellungsstatus versetzt werden.</p> <p>Sie können mit der Eigenschaft 'commandMessagePriority' auch die Priorität von internen Empfangsbestätigungsnachrichten und von Nachrichten, für die eine Empfangsbestätigung erwartet wird, festlegen.</p> <p>Um den internen Managed File Transfer-Nachrichten eine höhere Priorität als neuen Übertragungsanforderungen zuzuordnen, müssen Sie diese Eigenschaft auf einen Wert zwischen 1 (niedrigste Priorität) und 9 (höchste Priorität) setzen.</p> <p>Die Eigenschaft 'commandMessagePriority' hat den Standardwert 8. Wenn also das IBM MQ-Attribut DEFPRTY (Standardpriorität) in einer Agentenbefehlswarteschlange kleiner-gleich 7 ist, haben interne Vereinbarungsnachrichten Priorität vor neuen Übertragungsanforderungen. Wenn der Wert des Attributs DEFPRTY auf 8 oder 9 gesetzt ist, müssen Sie DEFPRTY oder die Eigenschaft 'commandMessagePriority' ändern, damit die Wirksamkeit der Eigenschaft 'commandMessagePriority' beibehalten bleibt.</p>	<p>Der Standardwert für IBM MQ 9.0.0.0 und höher ist 8.</p> <p>Bei früheren Releases und vor APAR IT06213 ist der Standardwert die Konstante MQPRI_PRIORITY_AS_Q_DEF, die einen Wert von -1 hat.</p>

Tabelle 33. Grundlegende Eigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
defaultProperties	Der Name des Standardsatzes an Konfigurationsoptionen. Bei diesem Wert handelt es sich um den Namen eines Verzeichnisses im Konfigurationsverzeichnis, in dem Verzeichnisse und Eigenschaftendateien mit Konfigurationsinformationen enthalten sind.	Kein Standardwert
enableFunctionalFixPack	<p>Die zu aktivierende Funktionsstufe des Fixpacks. Neue Funktionen, die in einem Fixpack enthalten sind, werden standardmäßig nicht aktiviert. Setzen Sie diese Eigenschaft auf eine Versions-ID, um die neuen Funktionen zu aktivieren, die für die betreffende Version verfügbar sind.</p> <p>Sie können die Versions-ID mit Punkten oder ohne Punkte (.) angeben. Wenn Sie beispielsweise die mit IBM MQ 8.0.0 Fix Pack 2 verfügbare Funktion verwenden möchten, setzen Sie diese Eigenschaft auf 8002 oder 8.0.0.2.</p>	Kein Standardwert




Tabelle 33. Grundlegende Eigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
messagePublicationFormat	<p>Ermöglicht Ihnen die Angabe des Formats zur Veröffentlichung von Nachrichten, das von MFT-Agenten für die zugehörigen XML-Statusnachrichten verwendet wird. Für diese Eigenschaft sind die folgenden Werte möglich:</p> <p><b>messagePublicationFormat=mixed</b> Nachrichten werden ohne ein MQMD-Format (MQFMT_NONE) veröffentlicht, mit Ausnahme der in der Themenstruktur '/LOG' veröffentlichten Nachrichten, welche im MQMD-Format MQFMT_STRING veröffentlicht werden.</p> <p><b>messagePublicationFormat=MQFMT_NONE</b> Nachrichten werden ohne ein MQMD-Format veröffentlicht.</p> <p><b>messagePublicationFormat=MQFMT_STRING</b> Nachrichten werden in einem Zeichenfolgeformat veröffentlicht.</p> <p>Vor IBM MQ 8.0 veröffentlichten MFT -Agenten XML-Statusnachrichten im SYSTEM.FTE in einem Zeichenfolgeformat (MQFMT_STRING) Sofern möglich, müssen Anwendungen, die zuvor IBM WebSphere MQ 7.5 verwendet haben, für die Verarbeitung von Nachrichten im Format der IBM MQ 8.0 oder höher aktualisiert werden. Wenn eine Anwendung nicht geändert werden kann, legen Sie die Eigenschaft 'messagePublicationFormat' auf MQFMT_STRING, um zum Verhalten der IBM WebSphere MQ 7.5 zurückzukehren.</p>	messagePublicationFormat=mixed

▶ z/OS
▶ z/OS
**z/OS-spezifische Eigenschaften:**



Tabelle 33. Grundlegende Eigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
  <b>productId</b>	<p>Produkttyp für den die MFT-Nutzung erfasst werden soll:</p> <ul style="list-style-type: none"> <li>• Eigenständiges Managed File Transfer-Produkt. (MFT ist die <b>productID</b>).</li> <li>• Komponente eines IBM MQ Advanced-Produkts. (ADVANCED ist die <b>productID</b>).</li> <li>• Komponente eines IBM MQ Advanced for z/OS Value Unit Edition-Produkts. (ADVANCEDVUE ist die <b>productID</b>).</li> </ul> <p>Weitere Informationen zur Aufzeichnung der Produktnutzung finden Sie unter <a href="#">Produktinformationen melden</a>.</p> <p> Diese Eigenschaft wird unter <a href="#">Multiplatforms</a> ignoriert.</p>	MFT

Der folgende Text ist ein Beispiel für den Inhalt einer Datei `installation.properties`.

```
defaultProperties=ERIS
```

ERIS ist der Name eines Verzeichnisses, das sich in demselben Verzeichnis wie die Datei `installation.properties` befindet. Das Verzeichnis ERIS enthält Verzeichnisse und Eigenschaftendateien, die eine Gruppe von Konfigurationsoptionen beschreiben.

#### Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

#### Zugehörige Verweise

[fteChangeDefaultConfigurationOptions](#)

## Die MFT-Datei 'coordination.properties'

Die Datei `coordination.properties` gibt die Verbindungsdetails für den Koordinations-WS-Manager an. Da mehrere Managed File Transfer-Installationen denselben Koordinationswarteschlangenmanager gemeinsam nutzen können, können Sie einen symbolischen Link zu einer gemeinsamen `coordination.properties`-Datei auf einem gemeinsam genutzten Laufwerk verwenden.

Die Datei `coordination.properties` wird vom Installationsprogramm oder über den Befehl **fteSetupCoordination** erstellt. Wenn Sie die grundlegenden Eigenschaften des Koordinationswarteschlangenmanagers in dieser Datei ändern möchten, verwenden Sie hierfür den Befehl **fteSetupCoordination** mit dem Flag **-f**. Um erweiterte Eigenschaften des Koordinationswarteschlangenmanagers zu ändern bzw. hinzuzufügen, muss die Datei in einem Texteditor bearbeitet werden.

Die Datei `coordination.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name`.

Die Datei `coordination.properties` enthält die folgenden Werte:

<i>Tabelle 34. Eigenschaften des Koordinationswarteschlangenmanagers</i>		
<b>Eigenschaftsname</b>	<b>Beschreibung</b>	<b>Standardwert</b>
coordinationQMgr	Name des Koordinationswarteschlangenmanagers.	Kein Standardwert
coordinationQMgrHost	Hostname oder IP-Adresse des Koordinationswarteschlangenmanagers.	Kein Standardwert
coordinationQMgrPort	Die für Clientverbindungen zum Koordinationswarteschlangenmanager verwendete Portnummer.	1414
coordinationQMgrChannel	Der SVRCONN-Kanalname, der zur Verbindung zum Koordinations-Warteschlangenmanager verwendet wird.	SYSTEM.DEF.SVRCONN

Wenn Sie keinen Wert für die Eigenschaft 'coordinationQMgrHost' angeben, wird standardmäßig der Bindungsmodus verwendet.

Wenn Sie für die Eigenschaft coordinationQMgrHost einen Wert angeben, nicht jedoch für die Eigenschaften von coordinationQMgrPort und coordinationQMgrChannel, so werden standardmäßig Portnummer 1414 und Kanal SYSTEM.DEF.SVRCONN verwendet.

<i>Tabelle 35. Erweiterte Eigenschaften für den Koordinationswarteschlangenmanager</i>		
<b>Eigenschaftsname</b>	<b>Beschreibung</b>	<b>Standardwert</b>
<b>Agenteneigenschaften:</b>		

Tabelle 35. Erweiterte Eigenschaften für den Koordinationswarteschlangenmanager (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
agentStatusJitterTolerance	<p>Die Höchstdauer, die die Veröffentlichung einer Nachricht über den Agentenstatus verzögert werden kann, bevor die Nachricht als überfällig betrachtet wird. Dieser Wert wird in Millisekunden gemessen.</p> <p>Das Alter einer Statusnachricht basiert auf dem Zeitpunkt, zu dem sie im Koordinationswarteschlangenmanager veröffentlicht wurde. Die Nachricht wird vom Agenten jedoch einige Zeit, bevor sie am Koordinationswarteschlangenmanager empfangen wird, ausgegeben, um der Dauer der Übertragung im IBM MQ-Netz Rechnung zu tragen. Dauert dieser Durchlauf immer gleich lange, werden Nachrichten, die mit einem Abstand von 60 Sekunden erstellt wurden, ungeachtet der tatsächlichen Übertragungsdauer mit einem Abstand von 60 Sekunden veröffentlicht. Wenn die Dauer des Transit jedoch zwischen Nachrichten variiert, werden sie möglicherweise in Intervallen von 60 Sekunden erstellt, aber in Intervallen von beispielsweise 61, 59, 58 und 62 Sekunden veröffentlicht. Die maximale Abweichung von 60 (in diesem Beispiel 2 Sekunden) wird als Jitter bezeichnet. Diese Eigenschaft legt die maximale Verzögerung aufgrund des Jitters vor der Behandlung der Nachricht als überfällig fest.</p>	3000
<b>Codepageeigenschaften</b>		

Tabelle 35. Erweiterte Eigenschaften für den Koordinationswarteschlangenmanager (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
coordinationCcsid	Die Codepage, mit der die Befehle eine Verbindung zum Koordinationswarteschlangenmanager herstellen. Diese Codepage wird auch für alle Veröffentlichungen des Agenten auf dem Koordinationswarteschlangenmanager verwendet. Wenn Sie einen Wert für die Eigenschaft 'coordinationCcsid' angeben, müssen Sie auch einen Wert für 'coordinationCcsidName' angeben.	1208
coordinationCcsidName	Die Java-Darstellung der ID des codierten Zeichensatzes für die Koordination (coordinationCcsid). Wenn Sie einen Wert für die Eigenschaft 'coordinationCcsidName' angeben, müssen Sie auch einen Wert für 'coordinationCcsid' angeben.	UTF8
<b>Verbindungseigenschaften:</b>		
javaLibraryPath	Wenn eine Verbindung zu einem Warteschlangenmanager im Bindungsmodus hergestellt wird, muss Managed File Transfer auf die Bindungsbibliotheken von IBM MQ Java zugreifen können. Managed File Transfer sucht die Bindungsbibliotheken standardmäßig an der durch IBM MQ definierten Standardposition. Wenn sich die Bindungsbibliotheken an einer anderen Position befinden, geben Sie mit dieser Eigenschaft die Position der Bindungsbibliotheken an.	MQ_INSTALLATION_PATH/java/lib
<b>Eigenschaften für Multi-Instanz-Warteschlangen:</b>		
coordinationQMgrStandby	Der Hostname und die Portnummer für Clientverbindungen (im IBM MQ CON-NAME-Format), und zwar für die Standby-Instanz eines Multi-Instanz-Koordinationswarteschlangenmanagers (definiert über die Eigenschaft 'coordinationQMgr'). Zum Beispiel, <i>host_name(port_number)</i>	Kein Standardwert

Tabelle 35. Erweiterte Eigenschaften für den Koordinationswarteschlangenmanager (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<b>Eigenschaften der Warteschlange:</b>		
dynamicQueuePrefix	<p>Diese Eigenschaft definiert das IBM MQ-Präfix für den Namen der temporären Warteschlange.</p> <p>Das Format der Eigenschaft 'dynamicQueuePrefix' entspricht dem Format des Feldes <b>DynamicQName</b> der MQOD-Struktur von IBM MQ. Weitere Informationen finden Sie im Abschnitt <a href="#">Dynamische Warteschlangen erstellen</a>.</p> <p>Diese Eigenschaft können Sie auch in der Datei <code>command.properties</code> definieren, wenn Sie ein bestimmtes IBM MQ-Präfix für temporäre Antwortwarteschlangen verwenden möchten, die durch Befehle generiert werden, von denen eine Antwort des Agenten benötigt wird.</p>	WMQFTE.*
modelQueueName	<p>Diese Eigenschaft definiert die IBM MQ-Modellwarteschlange, die zum Generieren einer temporären Warteschlange verwendet werden soll.</p> <p>Sie können diese Eigenschaft auch in der Datei '<code>command.properties</code>' definieren, wenn Sie eine bestimmte IBM MQ-Modellwarteschlange für temporäre Antwortwarteschlangen verwenden möchten, die über Befehle generiert werden, welche eine Antwort des Agenten erfordern. Weitere Informationen finden Sie unter <a href="#">„Die MFT-Datei 'command.properties'“</a> auf Seite 191.</p>	SYSTEM.DEFAULT.MODEL.QUEUE
<b>Sicherheitseigenschaften:</b>		

Tabelle 35. Erweiterte Eigenschaften für den Koordinationswarteschlangenmanager (Forts.)




Eigenschaftsname	Beschreibung	Standardwert
userIdForClientConnect	Die Benutzer-ID, die über Clientverbindungen an IBM MQ weitergegeben wird. Bei Angabe von <i>java</i> wird der von der JVM gemeldete Benutzername als Teil der IBM MQ-Verbindungsanforderung weitergegeben. Mögliche Werte für diese Eigenschaft sind None (Keine) oder <i>java</i> .	--
coordinationQMgrAuthenticationCredentialsFile	Der Pfad der Datei, die die MQ-Berechtigungsanzeige für die Verbindung mit dem Koordinationswarteschlangenmanager enthält.	<p> Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter <a href="#">MQMFTCredentials.xml</a> unter <a href="#">z/OS konfigurieren</a>.</p> <p> Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt <a href="#">MQMFTCredentials.xml konfigurieren</a>.</p> <p> Weitere Informationen zum Erstellen der Authentifizierungsberechtigungsdatei finden Sie unter <a href="#">MFT- und IBM MQ-Authentifizierungsnachweis</a>.</p>
<b>Subskriptionseigenschaften:</b>		

Tabelle 35. Erweiterte Eigenschaften für den Koordinationswarteschlangenmanager (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
coordinationSubscriptionTopic	<p>Verwenden Sie diese Eigenschaft, um ein anderes Thema als SYSTEM.FTE zu abonnieren, um Veröffentlichungen zum Status des IBM MQ -Netztes zu erhalten. Alle Tools werden weiterhin im SYSTEM.FTE , aber Sie können Ihre IBM MQ -Topologie ändern, um diese Veröffentlichungen je nach Inhalt an verschiedene Themen zu verteilen. Danach können Sie die Tools mit dieser Funktion so konfigurieren, dass sie eines dieser anderen Themen abonnieren.</p> <p>Für IBM WebSphere MQ 7.5 und spätere Fixpacks benötigen Sie einen vorläufigen Fix für APAR IC96850, damit diese Eigenschaft vom Plug-in IBM MQ Explorer und dem Befehl <b>fteListMonitors</b> erkannt wird.</p>	SYSTEM.FTE

Informationen zu SSL-Eigenschaften finden Sie unter „SSL-Eigenschaften für MFT“ auf Seite 233.

Das folgende Beispiel zeigt den Inhalt einer Datei `coordination.properties`:

```
coordinationQMGr=ERIS
coordinationQMGrHost=kuiper.example.com
coordinationQMGrPort=2005
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
```

In diesem Beispiel ist ERIS der Name eines IBM MQ-Warteschlangenmanagers auf dem System `kuiper.example.com`. ERIS ist der Warteschlangenmanager, an den Managed File Transfer Protokollinformationen sendet.

### Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

### Zugehörige Verweise

[fteSetupCoordination](#)

[„SSL-Eigenschaften für MFT“ auf Seite 233](#)

Einige MFT-Eigenschaftendateien enthalten SSL-Eigenschaften. Mit SSL oder TLS können Sie in IBM MQ und Managed File Transfer unberechtigte Verbindungen zwischen Agenten und Warteschlangenmanagern verhindern und die Nachrichtenübertragungen zwischen Agenten und Warteschlangenmanagern verschlüsseln.

## Die MFT-Datei 'command.properties'

In der Datei `command.properties` ist der Befehlswarteschlangenmanager angegeben, zu dem eine Verbindung hergestellt werden muss, wenn Befehle ausgegeben werden. Außerdem enthält die Datei Informationen, die Managed File Transfer für den Kontakt zu diesem Warteschlangenmanager benötigt.

Die Datei `command.properties` wird vom Installationsprogramm oder über den Befehl **fteSetupCommands** erstellt. Wenn Sie die grundlegenden Eigenschaften des Befehlswarteschlangenmanagers in dieser Datei ändern möchten, verwenden Sie hierfür den Befehl **fteSetupCommands** mit dem Flag **-f**. Um

erweiterte Eigenschaften des Befehlswarteschlangenmanagers zu ändern bzw. hinzuzufügen, muss die Datei in einem Texteditor bearbeitet werden.

Bei einigen Managed File Transfer-Befehlen wird eine Verbindung zum Agentenwarteschlangenmanager oder zum Koordinationswarteschlangenmanager und nicht zum Befehlswarteschlangenmanager hergestellt. Im Abschnitt [Verbindung zwischen MFT-Befehlen und Warteschlangenmanager](#) erfahren Sie, welche Befehle zu welchen Warteschlangenmanagern Verbindungen herstellen.

Die Datei `command.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name`.

Die Datei `command.properties` enthält die folgenden Werte:

<i>Tabelle 36. Grundlegende Eigenschaften des Befehlswarteschlangenmanagers</i>		
<b>Eigenschaftsname</b>	<b>Beschreibung</b>	<b>Standardwert</b>
connectionQMGr	Der Name des für die Verbindung mit dem IBM MQ-Netz verwendeten Warteschlangenmanagers.	Kein Standardwert
connectionQMGrHost	Hostname oder IP-Adresse des Verbindungs-Warteschlangenmanagers.	Kein Standardwert
connectionQMGrPort	Die für die Verbindung zum Verbindungs-Warteschlangenmanager in Client Mode verwendete Portnummer.	1414
connectionQMGrChannel	Der SVRCONN-Kanalname, der zur Verbindung zum Verbindungs-Warteschlangenmanager verwendet wird.	SYSTEM.DEF.SVRCONN

Wenn Sie keinen Wert für die Eigenschaft 'connectionQMGrHost' angeben, wird standardmäßig der Bindungsmodus verwendet.

Wenn Sie einen Wert für die Eigenschaft von connectionQMGrHost angeben, jedoch nicht für die Eigenschaften von connectionQMGrPort und connectionQMGrChannel, werden standardmäßig die Portnummer 1414 und der Kanal SYSTEM.DEF.SVRCONN verwendet.

<i>Tabelle 37. Erweiterte Eigenschaften für den Befehlswarteschlangenmanager</i>		
<b>Eigenschaftsname</b>	<b>Beschreibung</b>	<b>Standardwert</b>
<b>Codepageeigenschaften</b>		
connectionCcsid	Die Codepage, mit der die Befehle eine Verbindung zum Befehlswarteschlangenmanager herstellen. Wenn Sie einen Wert für die Eigenschaft 'connectionCcsid' angeben, müssen Sie auch einen Wert für 'connectionCcsidName' angeben.	1208
connectionCcsidName	Die Java-Darstellung der ID des codierten Zeichensatzes für Verbindungen (connectionCcsid). Wenn Sie einen Wert für die Eigenschaft 'connectionCcsidName' angeben, müssen Sie auch einen Wert für 'connectionCcsid' angeben.	UTF8



Tabelle 37. Erweiterte Eigenschaften für den Befehlswarteschlangenmanager (Forts.)





Eigenschaftsname	Beschreibung	Standardwert
<b>Eigenschaften für Multi-Instanz-Warteschlangen:</b>		
connectionQMGrStandby	Der Hostname und die Portnummer für Clientverbindungen (im IBM MQ CONNAME-Format), und zwar für die Standby-Instanz eines Multi-Instanz-Befehlswarteschlangenmanagers (definiert über die Eigenschaft 'connectionQMGr'). Zum Beispiel, <i>host_name(port_number)</i>	Kein Standardwert
<b>Sicherheitseigenschaften:</b>		
userIdForClientConnect	Die Benutzer-ID, die über Clientverbindungen an IBM MQ weitergegeben wird. Bei Angabe von <i>java</i> wird der von der JVM gemeldete Benutzername als Teil der IBM MQ-Verbindungsanforderung weitergegeben. Mögliche Werte für diese Eigenschaft sind None (Keine) oder <i>java</i> .	--
connectionQMGrAuthenticationCredentialsFile	Der Pfad der Datei, die die MQ-Berechtigungsanzeige für die Verbindung mit dem Befehlswarteschlangenmanager enthält.	<p> Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter <a href="#">MQMFTCredentials.xml</a> unter <a href="#">z/OS konfigurieren</a>.</p> <p> Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt <a href="#">MQMFTCredentials.xml konfigurieren</a>.</p> <p> Weitere Informationen zum Erstellen der Authentifizierungsberechtigungsdatei finden Sie unter <a href="#">MFT- und IBM MQ-Authentifizierungsnachweis</a>.</p>
<b>Eigenschaften der Warteschlange:</b>		

Tabelle 37. Erweiterte Eigenschaften für den Befehlswarteschlangenmanager (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
dynamicQueuePrefix	<p>Bei Befehlen, die eine Antwort vom Agenten benötigen, definiert diese Eigenschaft das IBM MQ-Präfix, das für die Generierung des Namens der temporären Antwortwarteschlange verwendet werden soll.</p> <p>Das Format der Eigenschaft 'dynamicQueuePrefix' entspricht dem Format des Feldes <b>DynamicQName</b> der MQOD-Struktur von IBM MQ. Weitere Informationen finden Sie im Abschnitt <a href="#">Dynamische Warteschlangen erstellen</a>.</p> <p>Sie können diese Eigenschaft auch in der Datei <code>coordination.properties</code> definieren, wenn Sie für von WMQFTE generierte temporäre Warteschlangen ein bestimmtes IBM MQ-Präfix verwenden möchten.</p>	WMQFTE.*
modelQueueName	<p>Bei Befehlen, die eine Antwort vom Agenten benötigen, definiert diese Eigenschaft die IBM MQ-Modellwarteschlange, die für die Generierung der temporären Antwortwarteschlange verwendet werden soll.</p> <p>Sie können diese Eigenschaft auch in der Datei 'coordination.properties' definieren, wenn Sie für von WMQFTE generierte temporäre Warteschlangen eine bestimmte IBM MQ-Modellwarteschlange verwenden möchten. Weitere Informationen finden Sie unter „Die MFT-Datei 'coordination.properties'“ auf <a href="#">Seite 185</a>.</p>	SYSTEM.DEFAULT.MODEL.QUEUE
<b>Verbindungseigenschaften:</b>		

Tabelle 37. Erweiterte Eigenschaften für den Befehlswarteschlangenmanager (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
javaLibraryPath	Wenn eine Verbindung zu einem Warteschlangenmanager im Bindungsmodus hergestellt wird, muss Managed File Transfer auf die Bindungsbibliotheken von IBM MQ Java zugreifen können. Managed File Transfer sucht die Bindungsbibliotheken standardmäßig an der durch IBM MQ definierten Standardposition. Wenn sich die Bindungsbibliotheken an einer anderen Position befinden, geben Sie mit dieser Eigenschaft die Position der Bindungsbibliotheken an.	/opt/mqm/java/lib
 legacyXMLMessageMQMDFormat	<p>Durch Managed File Transfer-Befehle generierte XML-Nachrichten werden nun an eine Warteschlange mit einem leeren MQMD-Formatfeld gesendet. In früheren Produktversionen war das MQMD-Formatfeld auf MQSTR gesetzt (eine Textnachrichten-Zeichenfolge). Wenn diese Eigenschaft auf "true" gesetzt ist, werden die durch Managed File Transfer generierten XML-Nachrichten an eine Warteschlange gesendet, deren MQMD-Formatfeld auf MQSTR gesetzt ist.</p> <p>Wenn das MQMD-Formatfeld auf MQSTR gesetzt ist und das MQ-Netz Kanäle mit Datenkonvertierung enthält, besteht die Gefahr, dass die XML-Nachrichten für Managed File Transfer-Befehle beschädigt werden.</p>	false
<b>Agenteneigenschaften</b>		
failCleanAgentWithNoArguments	Diese Eigenschaft hat standardmäßig den Wert 'true', was bedeutet, dass die Ausführung des Befehls <b>fteCleanAgent</b> fehlschlägt, wenn nur der Parameter für den Agentennamen angegeben ist. Wird die Eigenschaft auf 'false' gesetzt und nur der Parameter für den Agentennamen angegeben, verhält sich der Befehl <b>fteCleanAgent</b> so, als wäre der Parameter <b>-all</b> angegeben.	true

Informationen zu SSL-Eigenschaften finden Sie unter „SSL-Eigenschaften für MFT“ auf Seite 233.

Das folgende Beispiel zeigt den Inhalt einer Datei `command.properties`:

```
connectionQMgr=PLUTO
connectionQMgrHost=kuiper.example.com
connectionQMgrPort=1930
connectionQMgrChannel=SYSTEM.DEF.SVRCONN
```

In diesem Beispiel ist PLUTO der Name eines IBM MQ-Warteschlangenmanagers auf dem System `kuiper.example.com`. PLUTO ist der Warteschlangenmanager, zu dem die Managed File Transfer-Befehle eine Verbindung herstellen.

### Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

### Zugehörige Verweise

[„Java-Systemeigenschaften für MFT“ auf Seite 232](#)

Einige Managed File Transfer-Befehls- und Agenteneigenschaften müssen als Java-Systemeigenschaften definiert werden, da sie die Konfiguration für ältere Funktionen bereitstellen, die die Mechanismen der Befehle und Agenteneigenschaften nicht unterstützen.

[„SSL-Eigenschaften für MFT“ auf Seite 233](#)

Einige MFT-Eigenschaftendateien enthalten SSL-Eigenschaften. Mit SSL oder TLS können Sie in IBM MQ und Managed File Transfer unberechtigte Verbindungen zwischen Agenten und Warteschlangenmanagern verhindern und die Nachrichtenübertragungen zwischen Agenten und Warteschlangenmanagern verschlüsseln.

[fteSetupCommands: MFT-Datei 'command.properties' erstellen](#)

[fteCleanAgent: MFT-Agenten bereinigen](#)

## Die MFT `agent.properties`-Datei

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens `agent.properties`, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei `agent.properties` angegeben sein.

Die Datei `agent.properties` wird vom Installationsprogramm oder mit dem Befehl **fteCreateAgent**, **fteCreateBridgeAgent** oder **fteCreateCDAgent** erstellt. Sie können jeden dieser Befehle mit dem Flag `-f` verwenden, um die grundlegenden Eigenschaften des Agentenwarteschlangenmanagers und die erweiterten Agenteneigenschaften zu ändern, die dem Typ des zu erstellenden Agenten zugeordnet sind. Zum Ändern oder Hinzufügen der erweiterten Agenteneigenschaften muss die Datei in einem Texteditor bearbeitet werden.

**Multi** Bei der Multiplattform-Version befindet sich die Datei `agent.properties` für einen Agenten im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name`.

**z/OS** Unter z/OS befindet sich die Datei `agent.properties` an der Position `$BFG_CONFIG variable/mqft/config/coordination_qmgr_name/agents/agent_name`.

Nach einer Änderung der Datei `agent.properties` ist ein Neustart des Agenten erforderlich, damit die Änderungen übernommen werden.

In einigen Managed File Transfer-Eigenschaften, die Datei- oder Verzeichnispositionen darstellen, können Sie Umgebungsvariablen verwenden. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt). Weitere Informationen finden Sie unter [„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 179](#).

## Grundlegende Agenteneigenschaften

Jede Datei `agent.properties` enthält die folgenden grundlegenden Agenteneigenschaften:

Tabelle 38. Grundlegende Agenteneigenschaften

Eigenschaftsname	Beschreibung	Standardwert
agentName	Der Name des Agenten. Der Agentenname muss den Bezeichnungskonventionen für IBM MQ-Objekte entsprechen. Weitere Informationen finden Sie unter „Konventionen zum Benennen von MFT-Objekten“ auf Seite 253.	Kein Standardwert
agentDesc	Die Beschreibung des Agenten, sofern Sie sich dazu entschließen, eine Beschreibung anzugeben.	Kein Standardwert
agentQMgr	Der Name des Agentenwarteschlangenmanagers.	Kein Standardwert
agentQMgrHost	Der Hostname oder die IP-Adresse des Agentenwarteschlangenmanagers.	Kein Standardwert
agentQMgrPort	Die Portnummer, die für Clientverbindungen zum Agentenwarteschlangenmanager verwendet wird.	1414
agentQMgrChannel	Der SVRCONN-Kanalname, der für die Verbindung zum Agentenwarteschlangenmanager verwendet wird.	SYSTEM.DEF.SVRCONN
agentType	Der Typ des Agenten: <ul style="list-style-type: none"> <li>• Standardmäßiger Nicht-Bridgeagent (STANDARD)</li> <li>• Protokollbridgeagent (BRIDGE)</li> <li>• Connect:Direct-Bridgeagent (CD_BRIDGE)</li> <li>• Integrierter Agent, wie von IBM Integration Bus verwendet (EMBEDDED)</li> <li>• In Sterling File Gateway integrierter Agent (SFG)</li> </ul>	STANDARD

Wenn Sie keinen Wert für die Eigenschaft 'agentQMgrHost' angeben, wird standardmäßig der Bindungsmodus verwendet.

Wenn Sie für die Eigenschaft 'agentQMgrHost' einen Wert angeben, nicht jedoch für die Eigenschaften 'agentQMgrPort' und 'agentQMgrChannel', werden standardmäßig Portnummer 1414 und Kanal SYSTEM.DEF.SVRCONN verwendet.

## Erweiterte Agenteneigenschaften

Managed File Transfer bietet auch erweiterte Agenteneigenschaften, mit denen Sie Agenten konfigurieren können. Wenn Sie eine der folgenden Eigenschaften verwenden möchten, bearbeiten Sie die Datei `agent.properties` manuell, um die erforderlichen erweiterten Eigenschaften hinzuzufügen. Bei Dateipfadangaben unter Windows ist darauf zu achten, dass der als Trennzeichen eingesetzte umgekehrte Schrägstrich (\) als doppelter umgekehrter Schrägstrich (\\), also als umgekehrter Schrägstrich mit Escapezeichen (\) angegeben wird. Ebenso können Sie als Trennzeichen einen einzelnen Schrägstrich (/) verwenden. Weitere Informationen zur Verwendung von Escapezeichen in Java-Eigenschaftendateien finden Sie in der Oracle-Dokumentation [Javadoc for the Properties class](#).

- [Eigenschaften für die Agentengröße](#)
- [Codepageeigenschaften](#)
- [Befehlseigenschaften](#)
- [Verbindungseigenschaften](#)
- [Connect:Direct-Bridgeeigenschaften](#)
- [Eigenschaften des Agenten für die Übertragung von Datei an Nachricht und von Nachricht an Datei](#)

- Allgemeine Agenteneigenschaften
- **V 9.1.4** Hochverfügbarkeitseigenschaften
- Ein-/Ausgabeeigenschaften
- Eigenschaften für die Unterstützung auf mehreren Kanälen
- Eigenschaften für Multi-Instanz-Warteschlangen
- Eigenschaften des Prozesscontrollers
- Protokollbridge-Eigenschaften
- Protokolleigenschaften des Protokollbridgeagenten
- Warteschlangeneigenschaften
- Eigenschaften für die Ressourcenüberwachung
- Eigenschaften für das Stammverzeichnis
- Scheduler-Eigenschaft
- Sicherheitseigenschaften
- Eigenschaften für das Zeitlimit
- **V 9.1.0** Eigenschaften für das Zeitlimit für die Übertragungswiederherstellung
- Trace- und Protokollierungseigenschaften
- Eigenschaften für die Übertragungsgrenzwerte
- Eigenschaften für die Benutzerexitroutine
- Eigenschaften für die IBM MQ-Clientkomprimierung
- **z/OS** z/OS-spezifische Eigenschaften
- Andere Eigenschaften

Informationen zu SSL-Eigenschaften finden Sie unter „SSL-Eigenschaften für MFT“ auf Seite 233.

Tabelle 39. Erweiterte Agenteneigenschaften		
Eigenschaftsname	Beschreibung	Standardwert
<b>Größeneigenschaften für den Agenten:</b>		
agentCheckpointInterval	<p>Das Intervall in vollständigen Datenframes, in dem ein Prüfpunkt zu Wiederherstellungszwecken gesetzt wird. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>Wenn es ein Problem gibt, das eine Wiederherstellung der Übertragung verursacht, kann die Übertragung nur bis zu einer Prüfpunktgrenze wiederhergestellt werden. Das heißt, je höher dieser Wert (bei großen agentChunkSize-, agentWindowSize- und agentFrameSize-Werten), desto mehr Zeit benötigt der Agent zur Wiederherstellung von Übertragungen. Für zuverlässige Managed File Transfer-Netze, in denen Übertragungen selten in einen Wiederherstellungszustand übergehen, kann es vorteilhaft sein, diesen Wert zu erhöhen, um die Gesamtleistung zu verbessern.</p>	1
agentChunkSize	<p>Die Größe der einzelnen Transportblöcke (Chunkgröße) für den Transport der Dateidaten. Der Wert bezeichnet daher die maximale Größe der IBM MQ-Nachrichten, die zwischen den Quellen- und Zielagenten übertragen werden. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>Dieser Wert wird zwischen dem Quellen- und dem Zielagenten verhandelt. Der größere der beiden Werte wird verwendet. Wenn Sie den Wert dieser Eigenschaft ändern möchten, ändern Sie ihn sowohl auf dem Quellen- als auch auf dem Zielagenten.</p> <p>"agentChunkSize" ist ein ganzzahliger Wert. Beispiel: "agentChunkSize=10240" setzt die Chunkgröße auf 10 KB.</p>	262144 Bytes (entspricht 256 KB)

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
agentFrameSize	<p>Die Anzahl Fenster für den Übertragungsframe. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>In Netzen mit einer langen Latenzzeit kann eine Erhöhung dieses Werts die Gesamtleistung verbessern, da der Agent mehr Nachrichtenblöcke gleichzeitig aktiv verwalten muss.</p> <p>Der Wert dieser Eigenschaft, multipliziert mit agentWindowSize, multipliziert mit agentChunkSize, bezeichnet die Obergrenze der Speicherbelegung des Agenten für jede einzelne Übertragung. Beispiel: 262144-Byte-Blöcke x 10 x 5 = 12,5 MB für jede Übertragung.</p> <p>Hinweis: Wenn die Größe der in einer einzelnen Übertragung gesendeten Dateien kleiner als 12,5 MB ist, hat eine Erhöhung des Werts dieser Eigenschaft keine Auswirkung auf die Leistung der Übertragung.</p>	5
agentWindowSize	<p>Die Anzahl Blöcke für jedes Fenster. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.</p> <p>In Netzen mit einer langen Latenzzeit kann eine Erhöhung dieses Werts die Gesamtleistung verbessern. Der Grund dafür ist, dass der Agent mehr Nachrichtenblöcke gleichzeitig aktiv verwalten muss und die weniger Bestätigungsnachrichten an den Quellenagenten zurückgesendet werden.</p> <p>Der Wert dieser Eigenschaft, multipliziert mit agentFrameSize, multipliziert mit agentChunkSize, bezeichnet die Obergrenze der Speicherbelegung des Agenten für jede einzelne Übertragung und damit auch die Obergrenze des Volumens an IBM MQ-Nachrichtendaten in der Befehlswarteschlange des Zielagenten. Beispiel: 262144-Byte-Blöcke x 10 x 5 = Obergrenze von 12,5 MB für jede Übertragung.</p> <p>Hinweis: Wenn die Größe der in einer einzelnen Übertragung gesendeten Dateien kleiner als 12,5 MB ist, hat eine Erhöhung des Werts dieser Eigenschaft keine Auswirkung auf die Leistung der Übertragung.</p>	10
<b>Codepageeigenschaften</b>		
agentCcsid	Die Codepage, mit welcher der Agent eine Verbindung zum Agentenwarteschlangenmanager herstellt. Wenn Sie einen Wert für agentCcsid angeben, müssen Sie auch einen Wert für agentCcsidName angeben. Informationen zur Anzeige der bekannten Codepages für die JVM finden Sie unter der Beschreibung des Befehls <b>fteCreateBridgeAgent</b> unter dem Parameter <a href="#">-hsc</a> .	1208
agentCcsidName	Die Java-Darstellung der ID des codierten Zeichensatzes für die Koordination (agentCcsid). Wenn Sie einen Wert für agentCcsidName angeben, müssen Sie auch einen Wert für agentCcsid angeben.	UTF8
<b>Befehlseigenschaften:</b>		
maxCommandHandlerThreads	Steuert die Anzahl der für die erste Syntaxanalyse und die Verarbeitung von Übertragungsbefehlsnachrichten verfügbaren Threads. Aktive Threads benötigen eine Verbindung zum Warteschlangenmanager; wenn sie inaktiv sind, geben die Threads die Verbindung jedoch frei.	5
maxCommandOutput	Die Anzahl an Bytes, die maximal für die Befehlsausgabe gespeichert werden können. Diese Eigenschaft gilt für Befehle, die für einen verwalteten Aufruf angegeben werden, sowie für die Befehle 'preSource', 'postSource', 'preDestination' und 'postDestination' für eine verwaltete Übertragung. Dies begrenzt die Länge der Befehlsausgabe, die im Abschnitt SYSTEM.FTE in das Übertragungsprotokoll geschrieben wird.	10240
maxCommandRetries	Gibt an, wie oft der Agent maximal eine Befehlswiederholung zulässt. Diese Eigenschaft gilt für Befehle, die für einen verwalteten Aufruf angegeben werden, sowie für die Befehle 'preSource', 'postSource', 'preDestination' und 'postDestination' für eine verwaltete Übertragung.	9
maxCommandWait	Die vom Agenten zugelassene maximale Zeitspanne in Sekunden zwischen den Wiederholungen. Diese Eigenschaft gilt für Befehle, die für einen verwalteten Aufruf angegeben werden, sowie für die Befehle 'preSource', 'postSource', 'preDestination' und 'postDestination' für eine verwaltete Übertragung.	60

Tabella 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
immediateShutdownTimeout	<p>Beim sofortigen Herunterfahren eines Agenten können Sie mit dieser Eigenschaft die maximale Zeit in Sekunden angeben, die gewartet wird, bis die Übertragungen des Agenten abgeschlossen sind, bevor das Herunterfahren erzwungen wird.</p> <p><b>Anmerkung:</b> Setzen Sie diese Eigenschaft nicht auf einen Wert unter 10 Sekunden (Standardwert). Bei einer sofortigen Beendigung des Agenten wird ausreichend Zeit zum Beenden aller externen Prozesse benötigt. Wenn der Wert dieser Eigenschaft zu niedrig ist, bleiben einige Prozesse unter Umständen aktiv.</p> <p>Wenn diese Eigenschaft auf 0 gesetzt ist, wartet der Agent, bis alle ausstehenden Übertragungen gestoppt sind. Falls für diese Eigenschaft ein ungültiger Wert angegeben ist, wird der Standardwert verwendet.</p>	10
<b>Verbindungseigenschaften:</b>		
javaLibraryPath	<p>Wenn im Bindungsmodus eine Verbindung zu einem Warteschlangenmanager hergestellt wird, muss Managed File Transfer über Zugriff auf die IBM MQ Java-Bindungsbibliotheken verfügen. Managed File Transfer sucht die Bindungsbibliotheken standardmäßig an der von IBM MQ definierten Standardposition. Wenn sich die Bindungsbibliotheken an einer anderen Position befinden, geben Sie mit dieser Eigenschaft die Position der Bindungsbibliotheken an.</p>	--
<b>Connect:Direct-Bridge-Eigenschaften:</b>		
cdNode	<p>Diese Eigenschaft ist erforderlich, wenn Sie die Connect:Direct-Bridge verwenden möchten.</p> <p>Der Name des Connect:Direct-Knotens, der für die Übertragung von Nachrichten vom Connect:Direct-Bridgeagenten zu den Connect:Direct-Zielknoten verwendet werden soll. Dieser Knoten ist Teil der Connect:Direct-Bridge, also nicht der ferne Knoten, der Quelle oder Ziel der Übertragung ist. Weitere Informationen finden Sie im Abschnitt <a href="#">Die Connect:Direct-Bridge</a>.</p>	Kein Standardwert
cdNodeHost	<p>Der Hostname bzw. die IP-Adresse des Connect:Direct-Knotens, der für die Übertragung von Dateien vom Connect:Direct-Bridgeagenten zu den Zielknoten verwendet werden soll (der Connect:Direct-Bridgeknoten).</p> <p>Meist befindet sich der Connect:Direct-Bridgeknoten auf dem gleichen System wie der Connect:Direct-Bridgeagent. In diesen Fällen ist der Standardwert dieser Eigenschaft, nämlich die IP-Adresse des lokalen Systems, korrekt. Wenn Ihr System über mehrere IP-Adressen verfügt oder sich Ihr Connect:Direct-Bridgeknoten auf einem anderen System als dem System Ihres Connect:Direct-Bridgeagenten befindet und diese Systeme ein Dateisystem gemeinsam nutzen, geben Sie mit dieser Eigenschaft den richtigen Hostnamen für den Connect:Direct-Bridgeknoten an.</p> <p>Falls Sie die Eigenschaft 'cdNode' nicht definiert haben, wird diese Eigenschaft ignoriert.</p>	Der Hostname oder die IP-Adresse des lokalen Systems.
cdNodePort	<p>Die Portnummer des Connect:Direct-Bridgeknotens, die von Clientanwendungen für die Kommunikation mit dem Knoten verwendet wird. In der Connect:Direct-Produktdokumentation wird dieser Port als API-Port bezeichnet.</p> <p>Falls Sie die Eigenschaft 'cdNode' nicht definiert haben, wird diese Eigenschaft ignoriert.</p>	1363



Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
cdTmpDir	<p>Das Verzeichnis auf dem System, auf dem der Connect:Direct-Bridgeagent ausgeführt wird, in dem Dateien vorübergehend gespeichert werden, bevor sie an den Connect:Direct-Zielknoten übertragen werden.</p> <p>Diese Eigenschaft gibt den vollständigen Pfad des Verzeichnisses an, in dem Dateien vorübergehend gespeichert werden. Ist für 'cdTmpDir' beispielsweise /tmp festgelegt, werden die Dateien vorübergehend in das Verzeichnis /tmp gestellt.</p> <p>Der Connect:Direct-Bridgeagent und der Connect:Direct-Bridgeknoten müssen in der Lage sein, unter Verwendung desselben Pfadnamens auf das über diesen Parameter angegebene Verzeichnis zuzugreifen. Dies muss bei der Planung der Connect:Direct-Bridgeinstallation berücksichtigt werden. Erstellen Sie den Agenten nach Möglichkeit auf dem System, auf dem sich auch der Connect:Direct-Knoten für die Connect:Direct-Bridge befindet. Sind Agent und Knoten auf verschiedenen Systemen installiert, muss sich das Verzeichnis in einem gemeinsam genutzten Dateisystem befinden und es muss möglich sein, von beiden Systemen aus unter Verwendung desselben Pfadnamens auf dieses Verzeichnis zuzugreifen. Weitere Informationen zu den unterstützten Konfigurationen finden Sie im Abschnitt <a href="#">Die Connect:Direct-Bridge</a>.</p> <p>Falls Sie die Eigenschaft 'cdNode' nicht definiert haben, wird diese Eigenschaft ignoriert.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „<a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a>“ auf Seite 179.</p>	<p><code>value_of_java.io.tmpdir /cdbridge-agentName</code></p> <p>Unter Windows:</p> <p><code>value_of_java.io.tmpdir \cdbridge-agentName</code></p>
cdTrace	Gibt an, ob für Daten, die zwischen dem Connect:Direct-Bridgeagenten und seinem Connect:Direct-Knoten hin- und hergesendet werden, vom Agenten ein Trace durchgeführt wird. Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code> .	false
cdMaxConnectionRetries	Die maximale Anzahl an Connect:Direct-Verbindungsversuchen für eine Dateiübertragung, für die noch keine erfolgreiche Verbindung hergestellt werden konnte, bevor die Übertragung fehlschlägt.	-1 (die Anzahl der Versuche ist unbegrenzt)
cdMaxPartialWorkConnectionRetries	Die maximale Anzahl an Connect:Direct-Verbindungsversuchen für eine Dateiübertragung, für die bereits erfolgreich eine Verbindung hergestellt und Übertragungsarbeit erledigt wurde, bevor die Übertragung fehlschlägt.	-1 (die Anzahl der Versuche ist unbegrenzt)
cdMaxWaitForProcessEndStats	Gibt an, wie viele Millisekunden maximal gewartet werden soll, bis Informationen zur Beendigung des Connect:Direct-Prozesses nach Abschluss des Prozesses in den Connect:Direct-Knotenstatistikdaten erscheinen. Nach Ablauf dieses Zeitraums wird die Dateiübertragung als gescheitert angesehen. In der Regel sind die Informationen sofort verfügbar, unter bestimmten Fehlerbedingungen werden die Informationen jedoch nicht veröffentlicht. Unter diesen Bedingungen schlägt die Dateiübertragung nach Ablauf des durch diese Eigenschaft angegebenen Zeitraums fehl.	60000
cdAppName	Der Anwendungsname, den der Connect:Direct-Bridgeagent zur Verbindung mit dem zur Bridge gehörenden Connect:Direct-Knoten verwendet.	Managed File Transfer <i>aktuelle Version</i> . Dabei steht <i>aktuelle Version</i> für die Versionsnummer des Produkts.
cdNodeLocalPortRange	Der lokale Portbereich, der für Socketverbindungen zwischen dem Connect:Direct-Bridgeagenten und dem zur Bridge gehörenden Connect:Direct-Knoten verwendet werden kann. Das Format dieses Werts ist eine durch Kommas getrennte Liste von Werten oder Bereichen.	--
cdNodeProtocol	Das Protokoll, das der Connect:Direct-Bridgeagent zur Verbindung mit dem zur Bridge gehörenden Connect:Direct-Knoten verwendet. Folgende Werte sind gültig: <ul style="list-style-type: none"> <li>• TCPIP</li> <li>• SSL</li> <li>• TLS</li> </ul>	TCPIP
cdNodeKeystore	Der Pfad zum Keystore, der für die sichere Kommunikation zwischen dem Connect:Direct-Bridgeagenten und dem zur Bridge gehörenden Connect:Direct-Knoten verwendet wird.	--
	<p>Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „<a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a>“ auf Seite 179.</p>	

Tabella 39. Erweiterte Agenteneigenschaften (Forts.)







Eigenschaftsname	Beschreibung	Standardwert
cdNodeKeystoreType	Das Dateiformat des Keystores, der durch die Eigenschaft 'cdNodeKeystore' angegeben wird. Die folgenden Werte sind gültig: 'jks' und 'pkcs12'. Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert.	jks
cdNodeKeystoreCredentialsFile	Der Pfad der Datei, die die cdNodeKeystore-Berechtigungsnachweise enthält. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „ <a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a> “ auf Seite 179.	 Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter <a href="#">z/OS konfigurieren</a> .  Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt MQMFTCredentials.xml konfigurieren.  Weitere Informationen zum Erstellen der Authentifizierungsberechtigungsdatei finden Sie unter MFT- und IBM MQ-Authentifizierungsnachweis.
cdNodeTruststore	Der Pfad zum Truststore, der für die sichere Kommunikation zwischen dem Connect:Direct-Bridgeagenten und dem zur Bridge gehörenden Connect:Direct-Knoten verwendet wird. Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „ <a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a> “ auf Seite 179.	--
cdNodeTruststoreType	Das Dateiformat des Truststores, der durch die Eigenschaft 'cdNodeTruststore' angegeben wird. Die folgenden Werte sind gültig: 'jks' und 'pkcs12'. Wenn die Eigenschaft 'cdNodeProtocol' auf 'SSL' oder 'TLS' gesetzt ist, wird diese Eigenschaft ignoriert.	jks
cdNodeTruststoreCredentialsFile	Der Pfad der Datei, die die cdNodeTruststore-Berechtigungsnachweise enthält. Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten. Weitere Informationen finden Sie unter „ <a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a> “ auf Seite 179.	 Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter MQMFTCredentials.xml unter <a href="#">z/OS konfigurieren</a> .  Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt MQMFTCredentials.xml konfigurieren.  Weitere Informationen zum Erstellen der Authentifizierungsberechtigungsdatei finden Sie unter MFT- und IBM MQ-Authentifizierungsnachweis.
logCDProcess	Die Protokollstufe der Connect:Direct-Prozessprotokollierung, die im Ereignisprotokoll des Agenten in der Datei output0.log aufgezeichnet wird. Mögliche Werte für diese Eigenschaft sind None (Keine), Failures (Fehler) oder All (Alle).	--
<b>Eigenschaften des Agenten für die Übertragung von Datei an Nachricht und Nachricht an Datei:</b>		
deleteTmpFileAfterRenameFailure	Wenn Sie diese Eigenschaft auf den Wert 'false' setzen, wird dadurch sichergestellt, dass temporäre Dateien nicht aus dem Ziel gelöscht werden, wenn die Umbenennungsoperation fehlschlägt. In diesem Fall bleiben die übertragenen Daten im Ziel in einer temporären Datei (.part). Sie können diese Datei später manuell umbenennen. Standardmäßig hat diese Eigenschaft den Wert 'true'. Diese Eigenschaft ist sowohl für Übertragungen von Nachrichten an Dateien als auch von Dateien an Dateien gültig.	true

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
enableQueueInputOutput	Standardmäßig kann der Agent als Teil der Übertragung keine Daten aus einer Quellenwarteschlange lesen oder in eine Zielwarteschlange schreiben. Wenn für diese Eigenschaft der Wert 'true' angegeben wird, kann der Agent Übertragungen von Dateien an Nachrichten und von Nachrichten an Dateien vornehmen. Mögliche Werte für diese Eigenschaft sind true oder false.	false
enableSystemQueueInputOutput	Gibt an, ob der Agent in oder aus IBM MQ-Systemwarteschlangen schreiben kann. Systemwarteschlangen ist das Qualifikationsmerkmal SYSTEM vorangestellt. <b>Anmerkung:</b> Systemwarteschlangen werden von IBM MQ, Managed File Transfer und anderen Anwendungen zur Übertragung wichtiger Informationen verwendet. Durch eine Änderung dieser Eigenschaft ermöglichen Sie dem Agenten den Zugriff auf diese Warteschlangen. Wenn Sie diese Eigenschaft auf 'true' setzen, sollten Sie die Warteschlangen, auf die der Agent zugreifen kann, mittels Benutzer-Sandboxing einschränken.	false
enableClusterQueueInputOutput	Gibt an, ob der Agent für IBM MQ-Clusterwarteschlangen über Lese- oder Schreibzugriff verfügt. <b>Anmerkung:</b> Sie müssen zusätzlich zur Eigenschaft 'enableQueueInputOutput' die Agenteneigenschaft 'enableClusterQueueInputOutput' angeben.	false
maxDelimiterMatchLength	Die maximale Anzahl an Zeichen, die mit dem regulären Java-Ausdruck abgeglichen werden kann, mit dem eine Textdatei als Teil einer Übertragung aus einer Datei in eine Nachricht in mehrere Nachrichten aufgeteilt wird.	5
maxInputOutputMessageLength	Die maximale Länge einer Nachricht in Byte, die ein Agent aus einer Quellenwarteschlange liest bzw. in eine Zielwarteschlange schreibt. Die Eigenschaft 'maxInputOutputMessageLength' des Quellenagenten einer Übertragung bestimmt, wie viele Bytes aus einer Nachricht in der Quellenwarteschlange gelesen werden können. Die Eigenschaft 'maxInputOutputMessageLength' des Zielagenten einer Übertragung bestimmt, wie viele Bytes in eine Nachricht in der Zielwarteschlange geschrieben werden können. Wenn die Länge einer Nachricht den Wert dieser Eigenschaft überschreitet, schlägt die Übertragung mit einem Fehler fehl. Diese Eigenschaft hat keine Auswirkung auf interne Managed File Transfer-Warteschlangen. Informationen zum Ändern dieser Eigenschaft finden Sie im Abschnitt <a href="#">Hinweise zur Definition der MQ-Attribute und MFT-Eigenschaften für die Nachrichtengröße</a> .	1048576
monitorGroupRetryLimit	Gibt an, wie oft ein Überwachungsprogramm maximal eine erneute Übertragung aus einer Nachricht in eine Datei auslöst, wenn die Nachrichtengruppe weiterhin in der Warteschlange verbleibt. Wie oft die Übertragung aus einer Nachricht in eine Datei ausgelöst wird, ist im MQMD-Rücksetzungszähler der ersten Nachricht in der Gruppe festgelegt.  Bei einem Neustart des Agenten löst das Überwachungsprogramm erneut eine Übertragung aus. Dies gilt selbst dann, wenn die Anzahl der ausgelösten Übertragungen den Wert der Eigenschaft 'monitorGroupRetryLimit' überschreitet. Wenn dieses Verhalten dazu führt, dass die Anzahl der ausgelösten Übertragungen den Wert von 'monitorGroupRetryLimit' überschreitet, schreibt der Agent einen Fehler in sein Ereignisprotokoll.  Wird für diese Eigenschaft der Wert -1 angegeben, löst das Überwachungsprogramm die Übertragung unbegrenzt oft erneut aus, solange die Auslöserbedingung nicht erfüllt ist.	10
<b>Allgemeine Agenteneigenschaften:</b>		
agentStatusPublishRateLimit	Die maximale Geschwindigkeit in Sekunden, mit der der Agent seinen Status aufgrund einer Änderung des Dateiübertragungsstatus neu veröffentlicht.  Wenn Sie für diese Eigenschaft einen zu niedrigen Wert festlegen, kann dies die Leistung des IBM MQ-Netzes negativ beeinflussen.	30
agentStatusPublishRateMin	Die Mindestrate in Sekunden, in der der Agent seinen Status veröffentlicht. Dieser Wert muss größer-gleich dem Wert der Eigenschaft 'agentStatusPublishRateLimit' sein.	300

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
enableMemoryAllocationChecking	<p>Gibt an, ob der Managed File Transfer Agent vor dem Akzeptieren einer Übertragung überprüfen soll, ob ausreichend Speicher für die Ausführung einer Übertragung verfügbar ist. Die Prüfung erfolgt sowohl auf dem Quellen- als auch dem Zielagenten. Reicht der Speicher nicht aus, wird die Übertragung abgelehnt.</p> <p>Bei der Berechnung des für die Übertragung erforderlichen Speicherplatzes wird von dem maximal erforderlichen Speicherplatz ausgegangen. Deshalb kann der Wert höher sein, als der tatsächlich von der Übertragung belegte Speicherplatz. Aus diesem Grund ist die Anzahl gleichzeitig ausführbarer Übertragungen möglicherweise geringer, wenn die Eigenschaft 'enableMemoryAllocationChecking' auf 'true' gesetzt wird. Deswegen sollten Sie die Eigenschaft nur dann auf 'true' setzen, wenn Managed File Transfer häufiger mit einem Fehler aufgrund abnormaler Speicherbedingungen fehlschlägt. Bei Übertragungen aus Dateien in Nachrichten und aus Nachrichten in Dateien mit umfangreichen Nachrichtengrößen ist die Belegung großer Speichermengen wahrscheinlich.</p>	false
enableDetailedReplyMessages	<p>Wenn diese Eigenschaft auf "true" gesetzt ist, enthalten Antworten auf gesteuerte Übertragungsanforderungen Detailinformationen zu den übertragenen Dateien. Die Detailinformationen und das Format sind identisch mit den im Übertragungsprotokoll in den Fortschrittsnachrichten, d. h. dem Element &lt;transferSet&gt;, veröffentlichten Angaben. Weitere Informationen finden Sie im Abschnitt <a href="#">Formate für Dateiübertragungsprotokollnachricht</a>.</p> <p>Die detaillierten Antwortinformationen sind nur enthalten, wenn die gesteuerte Übertragungsanforderung angibt, dass detaillierte Antwortinformationen erforderlich sind. Um diese Anforderung anzugeben, setzen Sie das Attribut <code>detailled</code> des Elements &lt;reply&gt; der managedTransfer-XML-Anforderungsnachricht, die an den Quellenagenten gesendet wird. Weitere Informationen finden Sie im Abschnitt <a href="#">Format für Dateiübertragungsanforderungsnachricht</a>.</p> <p>Für jede Übertragungsanforderung können mehrere Antwortnachrichten generiert werden. Diese Anzahl entspricht der Anzahl der Übertragungsprotokoll-Statusnachrichten für die Übertragung plus 1 (hierbei ist die erste Antwortnachricht eine einfache ACK-Antwort). Detaillierte Informationen sind in allen Nachrichten enthalten, mit Ausnahme der ACK-Antwortnachrichten; das Gesamttransferergebnis ist jedoch nur in der aktuellsten ausführlichen Antwortnachricht enthalten.</p>	true
enableUserMetadataOptions	<p>Gibt an, ob bekannte Schlüssel für benutzerdefinierte Metadaten in neuen Übertragungsanforderungen verwendet werden können, um mehr Übertragungsoptionen bereitzustellen. Diese bekannten Schlüssel beginnen immer mit dem Präfix <code>com.ibm.wmqfte.</code>. Daraus folgt, dass bei Angabe des Werts 'true' für die Eigenschaft 'enableUserMetadataOptions' Schlüssel mit diesem Präfix nicht für eine benutzerdefinierte Verwendung unterstützt werden. Wenn die Eigenschaft 'enableUserMetadataOptions' auf 'true' gesetzt ist, werden zurzeit folgende Schlüssel unterstützt:</p> <ul style="list-style-type: none"> <li>• <code>com.ibm.wmqfte.insertRecordLineSeparator</code></li> <li>• <code>com.ibm.wmqfte.newRecordOnLineSeparator</code></li> <li>• <code>com.ibm.wmqfte.convertLineSeparators</code></li> </ul> <p>Informationen zur Bedeutung dieser Schlüssel finden Sie unter <a href="#">fteCreate-Transfer</a>: Neue Dateiübertragung starten.</p> <p>Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code>.</p>	false

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
failTransferOnFirstFailure	<p>Ein Agent kann so konfiguriert werden, dass eine verwaltete Übertragung fehlschlägt, sobald ein Übertragungselement innerhalb der verwalteten Übertragung fehlschlägt.</p> <p>Zum Aktivieren dieses Features muss APAR IT03450 sowohl für den Quellenagenten als auch für den Zielagenten angewendet werden und die Eigenschaft failTransferOnFirstFailure muss in der Datei agent.properties des Quellenagenten auf true gesetzt sein. Im Zielagenten ist das Festlegen der Eigenschaft auf true optional.</p> <p>Wenn die Eigenschaft 'failTransferOnFirstFailure' auf true gesetzt ist, startet der Agent die Verarbeitung der verwalteten Übertragung wie gewohnt. Sobald jedoch ein Übertragungselement fehlschlägt, wird die verwaltete Übertragung als fehlgeschlagen gekennzeichnet und es werden keine weiteren Übertragungselemente verarbeitet. Mit Übertragungselementen, die vor dem Fehlschlagen der verwalteten Übertragung erfolgreich verarbeitet wurden, wird wie folgt verfahren:</p> <ul style="list-style-type: none"> <li>Die Quellendisposition für diese Übertragungselemente wird berücksichtigt. Wenn die Quellendisposition für das Übertragungselement beispielsweise auf delete (Löschen) gesetzt war, wird die Quelldatei gelöscht.</li> <li>Die festgeschriebenen Zieldateien bleiben im Zielsystem und werden nicht gelöscht.</li> </ul> <p>Wenn die Eigenschaft 'failTransferOnFirstFailure' nicht auf true gesetzt ist und eine verwaltete Dateiübertragung mehrere Dateien enthält und die Übertragung einer dieser Dateien fehlschlägt (beispielsweise, weil die Zieldatei bereits vorhanden ist und die Eigenschaft zum Überschreiben auf error (Fehler) gesetzt ist), wird der Quellenagent fortgesetzt und versucht, die verbliebenen Dateien aus der Anforderung zu übertragen.</p>	false
itemsPerProgressMessage	<p>Die Anzahl Dateien, die übertragen werden, bevor ein Agent seine nächste Fortschrittsprotokollnachricht veröffentlicht. Steuern Sie über diese Eigenschaft die Rate, in der während einer Übertragung Fortschrittsprotokollnachrichten an den Koordinationswarteschlangenmanager veröffentlicht werden.</p> <p>Der maximale Wert für diese Eigenschaft ist 1000.</p> <p><b>Anmerkung:</b> Zu den Fortschrittsnachrichten gehören Informationen zu jeder Datei, die seit der Veröffentlichung der letzten Fortschrittsnachricht übertragen wird. Bei einer Erhöhung dieses Werts nimmt die Größe der Fortschrittsnachrichten zu, was die Leistung beeinträchtigen kann.</p>	50
maxInlineFileSize	<p>Bei Übertragungen aus Dateien in Dateien oder aus Dateien in Nachrichten die maximale Dateigröße (in Bytes), die automatisch in die erste Übertragungsanforderungsnachricht eingeschlossen werden kann.</p> <p>Mit dieser Eigenschaft können Sie die Geschwindigkeit der Übertragungen verbessern, doch wenn der als Dateigröße angegebene Wert zu groß ist, kann die Leistung dadurch auch beeinträchtigt werden. Als Anfangswert wird für diese Eigenschaft eine Dateigröße von 100 KB vorgeschlagen, doch es wird dringend empfohlen, sorgfältig mit verschiedenen Werten zu experimentieren, bevor Sie die für Ihr System geeignetste Dateigröße festlegen.</p> <p>Diese Funktion ist standardmäßig inaktiviert, oder indem Sie die Eigenschaft <b>maxInlineFileSize</b> auf 0 setzen.</p>	0
<p><b>V 9.1.4</b> <b>V 9.1.4</b> <b>Hochverfügbarkeitseigenschaften:</b></p>		
<b>V 9.1.4</b> highlyAvailable	<p>Diese Eigenschaft wird beim Start des Agenten gelesen. Wenn als Wert true angegeben ist, wird der Agent im Hochverfügbarkeitsmodus gestartet. Wenn Sie die Eigenschaft nicht angeben oder den Wert auf 'false' setzen, wird der Agent als nicht hochverfügbarer Agent gestartet.</p>	false
<b>V 9.1.4</b> standbyPollInterval	<p>Diese Eigenschaft wird von der Standby-Instanz verwendet, um zu versuchen, die gemeinsam genutzte Warteschlange in bestimmten Intervallen zu öffnen. Die Versuche werden wiederholt, bis die Standby-Instanz die Warteschlange SYSTEM.FTE.HA.&lt;agent name&gt; öffnet oder durch den Befehl <b>fteStop-Agent</b> gestoppt wird.</p>	5 Sekunden
<b>V 9.1.4</b> standbyStatusDiscardTime	<p>Diese Eigenschaft legt fest, wie lange die aktive Instanz auf eine Statusveröffentlichung von einer Standby-Instanz wartet.</p> <p>Wenn auch nach dieser Wartezeit keine Veröffentlichung von einer Standby-Instanz empfangen wird, entfernt die aktive Instanz die Informationen zu der Standby-Instanz aus ihrer Liste der Standby-Instanzen.</p> <p>Der Standardwert ist doppelt so hoch wie der der Eigenschaft <b>standbyStatusPublishInterval</b>, sodass die aktive Instanz länger wartet, bevor sie die Standby-Instanz aus ihrer Liste entfernt.</p>	600 Sekunden

Tabella 39. Erweiterte Agenteneigenschaften (Forts.)

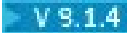




Eigenschaftsname	Beschreibung	Standardwert
 standbyStatusExpiry	Diese Eigenschaft legt die Ablaufzeit der Standby-Statusnachricht fest, die in die Befehlswarteschlange eines Agenten eingereicht wird. Die Nachricht läuft ab, wenn die aktive Instanz eines Agenten die Nachricht nicht verarbeitet.	30 Sekunden
 standbyStatusPublishInterval	Diese Eigenschaft legt fest, wie oft die Standby-Instanz ihren Status veröffentlicht.	300 Sekunden
<b>Ein-/Ausgabeeigenschaften:</b>		
doNotUseTempOutputFile	Standardmäßig schreibt der Agent Daten in eine temporäre Datei am Ziel und benennt diese temporäre Datei nach Abschluss der Dateiübertragung in den erforderlichen Dateinamen um. Wird dieser Wert auf 'true' gesetzt, schreibt der Agent direkt in die endgültige Zielfeile.  Auf z/OS-Systemen gilt dieses Verhalten nicht für sequenzielle Datasets, sondern für Member der partitionierten Datei. Der Wert dieser Eigenschaft für eine Übertragung wird vom Zielagenten definiert.	false
enableMandatoryLocking	Beim Zugriff auf normale Dateien verwendet Managed File Transfer für den Lesevorgang eine gemeinsame Sperre und für den Schreibvorgang eine exklusive Sperre.  Unter Windows stellt die Dateisperre lediglich eine Empfehlung dar. Wenn diese Eigenschaft auf 'true' gesetzt ist, erzwingt Managed File Transfer die Dateisperre. Unter Windows bedeutet dies, dass die Überwachung einer Datei, die durch eine andere Anwendung geöffnet ist, erst nach dem Schließen der Datei ausgelöst wird. Managed File Transfer-Übertragungen, an denen diese Datei beteiligt ist, schlagen fehl.  Auf UNIX-Plattformen wird die Dateisperre prozessübergreifend ausgeführt. Auf UNIX-Plattformen hat die Einstellung dieser Eigenschaft keine Auswirkung. Diese Eigenschaft gilt nur für normale Managed File Transfer-Agenten. Managed File Transfer unterstützt den Mechanismus für die Dateisperre auf einem Dateiserver nicht. Diese Eigenschaft funktioniert daher nicht für einen Protokollbridgeagenten, da ein Protokollbridgeagent eine Datei beim Übertragen einer Datei nicht auf einem Dateiserver sperrt. Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code> .	false
ioIdleThreadTimeout	Gibt in Millisekunden an, wie lange eine Inaktivität bei einem Ein-/Ausgabethread des Dateisystems vorliegen muss, bevor der Thread beendet wird.	10000
ioQueueDepth	Die Anzahl an Eingabe-/Ausgabeanforderungen, die maximal in die Warteschlange eingereicht werden können.	10
ioThreadPoolSize	Maximale Anzahl der verfügbaren Ein-/Ausgabethreads des Dateisystems. Für gewöhnlich verwendet jede Übertragung ihren eigenen Ein-/Ausgabethread des Dateisystems; wenn die Anzahl der gleichzeitig ablaufenden Übertragungen jedoch diesen Grenzwert überschreitet, werden die Ein-/Ausgabethreads des Dateisystems von mehreren Übertragungen geteilt. Ist davon auszugehen, dass regelmäßig mehr gleichzeitig ablaufende Übertragungen stattfinden werden als über den Wert 'ioThreadPoolSize' angegeben, kann unter Umständen eine Leistungsverbesserung erzielt werden, wenn dieser Wert erhöht wird, sodass es für jede Übertragung einen eigenen Ein-/Ausgabethread des Dateisystems gibt.	10
textReplacementCharacterSequence	Wenn bei einer Übertragung im Textmodus Datenbytes nicht von der Codepage der Quelle in die Codepage des Ziels konvertiert werden können, schlägt die Dateiübertragung standardmäßig fehl. Wenn Sie diese Eigenschaft festlegen, kann die Übertragung erfolgreich abgeschlossen werden, indem der angegebene Zeichenwert eingefügt wird. Dieser Eigenschaftswert ist ein einzelnes Zeichen. Normalerweise wird ein Fragezeichen (?) für Zeichen verwendet, die sich nicht zuordnen lassen. Verwenden Sie beispielsweise das Format 'textReplacementCharacterSequence=?'. Dabei ist das Fragezeichen (?) das Ersatzzeichen. Die Verwendung eines Leerzeichens als Ersatzzeichen ist nicht zulässig.	--
<b>Unterstützung auf mehreren Kanälen:</b>		

Tabella 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
agentMultipleChannelsEnabled	<p>Wenn Sie diese Eigenschaft auf <code>true</code> setzen, kann ein Managed File Transfer Agent Übertragungsdatennachrichten über mehrere IBM MQ -Kanäle senden. In einigen Szenarios kann die Einstellung dieser Eigenschaft die Leistung verbessern. Aktivieren Sie die Mehrkanalunterstützung jedoch nur, wenn eine erkennbare Leistungsverbesserung eintritt. Es werden nur Nachrichten, die in die Warteschlange 'SYSTEM.FTE.DATA.Zielagentenname' gestellt werden, über mehrere Kanäle gesendet. Das Verhalten aller anderen Nachrichten bleibt davon unberührt.</p> <p>Wenn Sie diese Eigenschaft auf <code>true</code> setzen, müssen Sie zur Aktivierung der Unterstützung mehrerer Kanäle auch die IBM MQ-Konfigurationsschritte ausführen, die in einem der folgenden Abschnitte beschrieben werden:</p> <ul style="list-style-type: none"> <li>„MFT-Agenten für mehrere Kanäle in einem Cluster konfigurieren“ auf Seite 219</li> <li>„MFT-Agenten für mehrere Kanäle konfigurieren: clusterunabhängig“ auf Seite 219</li> </ul> <p>Darüber hinaus müssen Sie die für einen Managed File Transfer-Agenten erforderliche IBM MQ-Standardkonfiguration ausführen (Details enthält der Abschnitt MFT für die erste Verwendung konfigurieren).</p> <p>Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code>.</p>	false
agentMessageBatchSize	<p>Wenn ein Quellenagent mit mehreren Kanälen konfiguriert wurde, sendet er Datennachrichten für eine Übertragung auf Umlaufbasis über jeden einzelnen Kanal. Diese Eigenschaft steuert die Anzahl der Nachrichten, die an jeweils einen Kanal gesendet werden.</p>	5
<b>Eigenschaften für Multi-Instanz-Warteschlangen:</b>		
agentQMGrStandby	<p>Der Hostname und die Portnummer für Clientverbindungen (im IBM MQ CON-NAME-Format), und zwar für die Standby-Instanz eines Agentenwarteschlangenmanagers mit mehreren Instanzen (definiert über die Eigenschaft 'connectionQMGr'). Zum Beispiel, <code>host_name(port_number)</code></p> <p>The agent attempts to connect to the standby queue manager when it detects a connection broken error, for example, MQRC 2009. Sobald der Agent mit dem Standby-WS-Manager verbunden wird, bleibt der Agent verbunden, bis der Standby-WS-Manager nicht mehr verfügbar ist.</p>	Kein Standardwert
<b>Eigenschaften des Prozesscontrollers:</b>		
agentQMGrRetryInterval	<p>Das Zeitintervall in Sekunden, in dem der Prozesscontroller des Agenten prüft, ob der Warteschlangenmanager verfügbar ist.</p>	30
maxRestartCount	<p>Die maximale Anzahl Neustarts, die innerhalb des Zeitintervalls, das durch den Wert der Eigenschaft 'maxRestartInterval' angegeben ist, ausführbar sind. Wird dieser Wert überschritten, führt der Prozesscontroller des Agenten keinen Neustart des Agenten mehr durch; stattdessen führt er eine Aktion aus, die auf dem Wert der Eigenschaft 'maxRestartDelay' basiert.</p>	4
maxRestartInterval	<p>Der Zeitraum in Sekunden, über den der Prozesscontroller des Agenten die Agentenneustarts zählt. Überschreitet die Anzahl der Neustarts innerhalb dieses Zeitraums den Wert der Eigenschaft 'maxRestartCount', führt der Prozesscontroller des Agenten keine Agentenneustarts mehr durch. Stattdessen führt der Prozesscontroller des Agenten eine Aktion aus, die auf dem Wert der Eigenschaft 'maxRestartDelay' basiert.</p>	120
maxRestartDelay	<p>Gibt das Verhalten des Agentenprozesscontrollers an, wenn die Anzahl der Agentenneustarts den Wert der Eigenschaften 'maxRestartCount' und 'maxRestartInterval' übersteigt. Wenn Sie einen Wert kleiner-gleich null angeben, wird der Prozesscontroller des Agenten gestoppt. Wenn Sie einen Wert größer als null angeben, ist dies die Zeit in Sekunden, die gewartet wird, bevor die im Prozesscontroller des Agenten enthaltenen Protokolldaten zu Neustarts zurückgesetzt werden und der Agent erneut gestartet wird.</p>	-1
<b>Protokollbridge-Eigenschaften:</b>		
protocolBridgeCredentialConfiguration	<p>Der Wert dieser Eigenschaft wird als Zeichenfolge an die Methode 'initialize()' der Exitklassen übergeben, die durch die Eigenschaft 'protocolBridgeCredentialExitClasses' angegeben sind.</p>	null
protocolBridgeCredentialExitClasses	<p>Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine für Protokollbridgeberechtigungs-nachweise implementieren. Weitere Informationen finden Sie im Abschnitt <a href="#">Berechtigungs-nachweise für einen Dateiserver mittels Exitklassen zuordnen</a>.</p>	Kein Standardwert.

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)



Eigenschaftsname	Beschreibung	Standardwert
protocolBridgeDataTimeout	Die Zeit in Millisekunden, die der Protokollbridgeagent wartet, bis entweder eine Datenverbindung mit einem FTP-Server zustande kommt oder bis Daten von einem FTP-Server über eine bereits vorhandene Verbindung eingehen. Der Wert 0 bedeutet kein Zeitlimit. Bei Verstreichen des Zeitlimits schließt der Protokollbridgeagent alle bestehenden Datenverbindungen mit dem FTP-Server und versucht die aktuelle Übertragung über eine neue Datenverbindung fortzusetzen. Lässt sich diese neue Datenverbindung nicht herstellen, schlägt auch die aktuelle Übertragung fehl.	0
protocolBridgeLogoutBeforeDisconnect	Gibt an, ob der Protokollbridgeagent den Benutzer vom Dateiserver abmeldet, bevor er die FTP-Sitzung abschließt und die Verbindung unterbricht. Wenn Sie diese Eigenschaft auf <code>true</code> setzen, gibt der Protokollbridgeagent einen <b>QUIT</b> -Befehl für die FTP-Sitzung an den Dateiserver aus.	false
protocolBridgePropertiesConfiguration	Der Wert dieser Eigenschaft wird an die Methode 'initialize()' der Exitklassen, die durch die Eigenschaft 'protocolBridgeServerPropertiesExitClasses' angegeben sind, als eine der Bridgeeigenschaften übergeben.	Kein Standardwert
protocolBridgePropertiesExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine für Servereigenschaften der Protokollbridge implementieren. Weitere Information finden Sie im Abschnitt <a href="#">ProtocolBridgePropertiesExit2: Protokolldateiservereigenschaften nachschlagen</a> .	Kein Standardwert
<b>Protokollierungseigenschaften des Protokollbridgeagenten:</b>		
agentLog	Schlüssel/Wert-Paar-Komponente und Operation zur Aktivierung bzw. Inaktivierung der Protokollierung von FTP-Befehlen und -Antworten zwischen dem Protokollbridgeagenten und den FTP-/SFTP-/FTPS-Dateiservern.  For example: <b>agentLog=on</b> Aktiviert die Protokollierung für alle Komponenten. <b>agentLog=off</b> Inaktiviert die Protokollierung für alle Komponenten. <b>agentLog=ftp=on, sftp=on, ftps=off</b> Aktiviert die Protokollierung für FTP und SFTP, inaktiviert sie für FTPS.	Kein Standardwert
agentLogFileSize	Gibt die maximale Größe einer Aufzeichnungsdatei in Megabytes an. Standard wie Standarddateigröße für regulären Trace.	20
agentLogFiles	Gibt an, wie viele Aufzeichnungsdateien maximal aufbewahrt werden, bevor die älteste Datei gelöscht wird.	10
agentLogFilter	Erfasst standardmäßig die Kommunikation mit allen FTP-Servern, mit denen der Agent eine Verbindung herstellt.  For example: <ul style="list-style-type: none"> <li>Filter für Host/IP-Adresse:  <pre>host=ftpprod.ibm.com, ftp2.ibm.com host=9.182.*</pre> </li> <li>Filter auf Basis der Metadaten  <pre>metadata="outbound files to xyz corp"</pre> </li> </ul>	*
<b>Eigenschaften der Warteschlange:</b>		
 dynamicQueue-Prefix	Diese Eigenschaft definiert das Präfix, das beim Erstellen einer temporären dynamischen Warteschlange verwendet wird.	WMQFTE.*
 modelQueueName	Diese Eigenschaft definiert den Namen der Modulwarteschlange, die beim Erstellen einer temporären dynamischen Warteschlange verwendet wird.	SYSTEM.DEFAULT.MODEL.QUEUE
publicationMDUser	Die MQMD-Benutzer-ID, die den Nachrichten zugeordnet wird, welche zur Veröffentlichung durch den Koordinationswarteschlangenmanager gesendet werden. Wenn Sie diese Eigenschaft nicht festlegen, wird die MQMD-Benutzer-ID auf Basis der IBM MQ-Regeln für die Einstellung von MQMD-Benutzer-IDs festgelegt.	Kein Standardwert
<b>Eigenschaften für die Ressourcenüberwachung:</b>		



Tabella 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
monitorFilepathPlatformSeparator	Gibt an, ob in der Variablen <code>\$FILEPATH</code> plattformspezifische Pfadtrennzeichen verwendet werden sollen. Der Wert <code>true</code> bedeutet, dass plattformspezifische Pfadtrennzeichen verwendet werden. Bei Angabe von <code>false</code> wird auf allen Plattformen der UNIX-Schrägstrich ( <code>/</code> ) als Pfadtrennzeichen verwendet.	true
monitorMaxResourcesInPoll	Gibt die maximale Anzahl an überwachten Ressourcen an, die in den einzelnen Abfrageintervallen ausgelöst werden sollen. Wenn Sie beispielsweise das Überwachungsmuster <code>*.txt</code> sowie ein Abfrageintervall von zehn Sekunden angeben und die Eigenschaft 'monitorMaxResourcesInPoll' auf '10' setzen, beschränkt die Eigenschaft 'monitorMaxResourcesInPoll' den Agenten dahingehend, dass er in jedem Abfrageintervall höchstens für zehn Übereinstimmungen eine Auslösung vornimmt. Übereinstimmende Ressourcen, die über den Grenzwert 10 hinausgehen, werden in späteren Abfrageintervallen ausgelöst.  Sie können die Eigenschaft 'monitorMaxResourcesInPoll' außerdem mit einem entsprechenden Wert für den Parameter <code>-bs</code> im Befehl <code>fteCreateMonitor</code> kombinieren, um beispielsweise jedes Abfrageintervall auf die Auslösung von nur einer Übertragung zu beschränken.  Ein Wert kleiner-gleich null bedeutet, dass die Anzahl der Überwachungsressourcen, die in einem Abfrageintervall ausgelöst werden, unbegrenzt ist.	-1
monitorReportTriggerFail	Gibt an, ob Fehlerbedingungen in der Umgebung oder in der Konfiguration, die bei der Überwachung erkannt werden, als Protokollnachricht an den Abschnitt <code>SYSTEM.FTE</code> gemeldet werden. Bei Angabe von <code>true</code> werden Nachrichten protokolliert, bei Angabe von <code>false</code> werden keine Nachrichten protokolliert.	true
monitorReportTriggerNotSatisfied	Gibt an, ob bei einem nicht erfüllten Auslöser eine Protokollnachricht an den Abschnitt <code>SYSTEM.FTE</code> gesendet wird, in dem sich die Details befinden. Bei Angabe von <code>true</code> werden Nachrichten protokolliert, bei Angabe von <code>false</code> werden keine Nachrichten protokolliert.	false
monitorReportTriggerSatisfied	Gibt an, ob bei einem erfüllten Auslöser eine Protokollnachricht an den Abschnitt <code>SYSTEM.FTE</code> gesendet wird, in dem sich die Details befinden. Bei Angabe von <code>true</code> werden Nachrichten protokolliert, bei Angabe von <code>false</code> werden keine Nachrichten protokolliert.	false
monitorSilenceOnTriggerFailure	Gibt an, nach wie vielen aufeinanderfolgenden Fehlern des Ressourcenüberwachungsauslösers die Fehler nicht mehr gemeldet werden.	5
monitorStopOnInternalFailure	Gibt an, nach wie vielen aufeinanderfolgenden internen FFDC-Bedingungen der Ressourcenüberwachung der Status der Überwachung in 'gestoppt' geändert wird.	10
<b>Eigenschaften des Stammverzeichnisses:</b>		

Tabella 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
commandPath	<p>Gibt die Pfade an, über die Befehle aufgerufen werden können:</p> <ul style="list-style-type: none"> <li>Über die Ant-Tasks <a href="#">fte:call</a> Ant, <a href="#">fte:filecopy</a> oder <a href="#">fte:filemove</a> des Agenten</li> <li>Unter Verwendung eines der unterstützten Managed File Transfer Agent-Befehls-XML-Schemas (beispielsweise 'managedCall' oder 'managedTransfer') in einer XML-Nachricht, die an einen Agenten übergeben wird.</li> </ul> <p>Informationen zur gültigen Syntax des Werts der Eigenschaft 'commandPath' finden Sie im Abschnitt <a href="#">MFT-Eigenschaft 'commandPath'</a>.</p> <p><b>Wichtig:</b> Gehen Sie beim Setzen dieser Eigenschaft sehr vorsichtig vor, da jeder Befehl in einem der angegebenen Befehlspfade (commandPaths) von jedem fernen Clientsystem aus aufgerufen werden kann, das in der Lage ist, Befehle an den Agenten zu senden. Daher gilt standardmäßig Folgendes, wenn Sie einen Befehlspfad angeben:</p> <ul style="list-style-type: none"> <li>Vorhandene Agentensandboxes werden vom Agenten bei dessen Start konfiguriert, sodass alle commandPath-Verzeichnisse automatisch zur Liste der Verzeichnisse hinzugefügt werden, die für Übertragungszugriffe gesperrt sind.</li> <li>Alle vorhandenen Benutzersandboxes werden beim Start des Agenten aktualisiert, sodass alle commandPath-Verzeichnisse (und ihre Unterverzeichnisse) als <code>&lt;exclude&gt;</code>-Elemente zu den Elementen <code>&lt;read&gt;</code> und <code>&lt;write&gt;</code> hinzugefügt werden.</li> <li>Wenn der Agent weder für die Verwendung einer Agentensandbox noch für Benutzersandboxes konfiguriert ist, wird beim Start des Agenten eine neue Agentensandbox erstellt, in der die commandPath-Verzeichnisse als gesperrte Verzeichnisse angegeben sind.</li> </ul> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „<a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a>“ auf Seite 179.</p> <p>Sie können die Eigenschaft <code>addCommandPathToSandbox</code> auf 'false' setzen, um dieses Standardverhalten aus Gründen der Kompatibilität mit folgenden Releases zu überschreiben:</p> <ul style="list-style-type: none"> <li>IBM WebSphere MQ File Transfer Edition.</li> <li>Die IBM WebSphere MQ 7.5.0 Fix Pack 1 Managed File Transfer-Komponente (oder früher).</li> <li>Der IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer-Komponente (oder später) in einer Installation, bei der die Installationseigenschaft 'enableFunctionalFixPack=7502' nicht gesetzt ist.</li> </ul> <p><b>Wichtig:</b> Es ist jedoch zu beachten, dass diese Überschreibung es einem Client ermöglicht, jeden beliebigen Befehl an das Agentensystem zu übertragen und aufzurufen. Deshalb sollte sie mit Vorsicht eingesetzt werden.</p>	Es können keine Befehle aufgerufen werden
addCommandPathToSandbox	<p>Gibt an, ob die mit der Eigenschaft <code>commandPath</code> angegebenen Verzeichnisse (und alle ihre Unterverzeichnisse) hinzugefügt werden sollen zu:</p> <ul style="list-style-type: none"> <li>den gesperrten Verzeichnissen für eine vorhandene Agentensandbox</li> <li>Die <code>&lt;exclude&gt;</code>-Elemente für die Elemente <code>&lt;read&gt;</code> und <code>&lt;write&gt;</code> für alle definierten Benutzersandboxes.</li> <li>einer neuen Agentensandbox, wenn ein Agent weder mit einer Agentensandbox noch mit einer Benutzersandbox konfiguriert wurde</li> </ul> <p>Dies stellt Kompatibilität mit folgenden Releases her:</p> <ul style="list-style-type: none"> <li>IBM WebSphere MQ File Transfer Edition.</li> <li>Die IBM WebSphere MQ 7.5.0 Fix Pack 1 Managed File Transfer-Komponente (oder früher).</li> <li>Der IBM WebSphere MQ 7.5.0 Fix Pack 2 Managed File Transfer-Komponente (oder später) in einer Installation, bei der die Installationseigenschaft 'enableFunctionalFixPack=7502' nicht gesetzt ist.</li> </ul> <p>Weitere Informationen finden Sie im Abschnitt <a href="#">MFT-Eigenschaft 'commandPath'</a>.</p>	True

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
additionalWildcardSandboxChecking	<p>Gibt an, ob für einen Agenten, der mit einer Benutzer- oder Agentensandbox konfiguriert wurde, zusätzliche Überprüfungen für Platzhalterübertragungen durchgeführt werden sollen, um die Positionen zu beschränken, an die und von denen der Agent Dateien übertragen kann.</p> <p>Ist diese Eigenschaft auf 'true' gesetzt, ist die zusätzliche Überprüfung aktiviert. Versucht eine Übertragungsanforderung, eine Position zu lesen, die außerhalb der definierten Sandbox für Dateiabgleich mit dem Platzhalter liegt, schlägt die Übertragung fehl. Wenn eine Übertragungsanforderung aus mehreren Übertragungen besteht und eine dieser Übertragungen fehlschlägt, weil sie versucht, eine Position außerhalb der Sandbox zu lesen, schlägt die gesamte Übertragung fehl. Schlägt die Überprüfung fehl, wird die Ursache dafür in einer Fehlermeldung angezeigt (siehe <a href="#">Zusätzliche Überprüfungen für Platzhalterübertragungen</a>).</p> <p>Wird die Eigenschaft weggelassen oder auf 'false' gesetzt, werden keine zusätzlichen Überprüfungen für Platzhalterübertragungen durchgeführt.</p>	--
sandboxRoot	<p>Gibt eine Reihe von Stammverzeichnispfaden an, die bei Aktivierung der Sandbox-Funktion berücksichtigt bzw. nicht berücksichtigt werden sollen. Informationen zu dieser Funktion finden Sie im Abschnitt <a href="#">Mit Sandboxes für den MFT-Agenten arbeiten</a>.</p> <p>Pfadnamen müssen mit einem plattformspezifischen Pfadtrennzeichen getrennt werden. Stellen Sie den Pfadangaben ein Ausrufezeichen (!) voran, um anzugeben, dass diese Pfade aus der Sandbox ausgeschlossen werden sollen. Dies ist hilfreich, wenn ein Unterverzeichnis in einem einbezogenen Stammverzeichnispfad ausgeschlossen werden soll.</p> <p>Die Eigenschaft 'sandboxRoot' wird auf Protokollbridgeagenten nicht unterstützt.</p> <p>Die Eigenschaften 'sandboxRoot' und 'userSandboxes' können nicht gemeinsam angegeben werden.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „<a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a>“ auf Seite 179.</p>	-- - keine Sandbox
transferRoot	<p>Standardmäßiges Stammverzeichnis für relative Pfade, die dem Agenten angegeben werden.</p> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „<a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a>“ auf Seite 179.</p>	Das Ausgangsverzeichnis für den Benutzer, der den Agentenprozess gestartet hat.
transferRootHLQ	<p>Standardmäßiges Qualifikationsmerkmal der höheren Ebene (Benutzer-ID) für nicht vollständig qualifizierende Datengruppen, die dem Agenten angegeben werden</p>	Der Benutzername des Benutzers, der den Agentenprozess gestartet hat.
userSandboxes	<p>Schränkt den Bereich des Dateisystems, in den und aus dem Dateien übertragen werden können, auf Grundlage des MQMD-Benutzernamens des Benutzers ein, der die Übertragung anfordert. Weitere Informationen finden Sie im Abschnitt <a href="#">Mit Sandboxes für MFT-Benutzer arbeiten</a>.</p> <p>Die Eigenschaft 'userSandboxes' wird auf Protokollbridgeagenten nicht unterstützt.</p> <p>Die Eigenschaften 'sandboxRoot' und 'userSandboxes' können nicht gemeinsam angegeben werden.</p>	false
<b>Scheduler-Eigenschaft:</b>		
maxSchedulerRunDelay	<p>Das maximale Intervall in Minuten, das der Agent abwartet, bis er prüft, ob geplante Übertragungen vorhanden sind. Wenn Sie diese Eigenschaft aktivieren möchten, geben Sie eine positive Ganzzahl an. Im Abschnitt <a href="#">Vorgehensweise</a>, wenn die geplante Dateiübertragung nicht oder verzögert ausgeführt wird finden Sie weitere Informationen darüber, ob diese Eigenschaft für Sie sinnvoll sein könnte.</p> <p>Da der Agent bei der Fälligkeit geplanter Übertragungsausführungen möglicherweise gerade einen Befehl aus seiner Befehlswarteschlange liest, kann sich der Start der geplanten Übertragungen unter Umständen weiter verzögern. In diesem Fall wird der Scheduler unmittelbar nach Abschluss des betreffenden Befehls ausgeführt.</p>	-1
<b>Sicherheitseigenschaften:</b>		

Tabella 39. Erweiterte Agenteneigenschaften (Forts.)




Eigenschaftsname	Beschreibung	Standardwert
agentQMGrAuthenticationCredentialsFile	Der Pfad der Datei, die die Berechtigungsnachweise für die Verbindung mit MQ enthält.	 Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter <a href="#">MQMFTCredentials.xml</a> unter <a href="#">z/OS konfigurieren</a> .  Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt <a href="#">MQMFTCredentials.xml konfigurieren</a> .  Weitere Informationen zum Erstellen der Authentifizierungsberechtigungsdatei finden Sie unter <a href="#">MFT- und IBM MQ-Authentifizierungsnachweis</a> .
authorityChecking	Legt fest, ob die im Abschnitt <a href="#">Benutzerberechtigungen für MFT-Agentenaktionen beschränken</a> beschriebenen Sicherheitsfunktionen aktiviert sind.  Die Berechtigung <code>inquire</code> ist eine erforderliche Berechtigung für alle Berechtigungswarteschlangen des Agenten.	false
logAuthorityChecks	Die Protokollstufe für die Berechtigungsprüfung, die im Ereignisprotokoll des Agenten in der Datei <code>output0.log</code> aufgezeichnet wird. Mögliche Werte für diese Eigenschaft sind <code>None</code> (Keine), <code>Failures</code> (Fehler) oder <code>All</code> (Alle).	--
userIdForClientConnect	Die Benutzer-ID, die über Clientverbindungen an IBM MQ weitergegeben wird. Bei Angabe von <code>java</code> wird der von der JVM gemeldete Benutzername als Teil der IBM MQ-Verbindungsanforderung weitergegeben. Mögliche Werte für diese Eigenschaft sind <code>None</code> (Keine) oder <code>java</code> .	--
<b>Eigenschaften für das Zeitlimit:</b>		
maxTransferNegotiationTime	Die maximale Zeit in Millisekunden, die eine Übertragung auf den Abschluss der Verhandlung durch den Zielagenten wartet. Wenn die Verhandlung nicht innerhalb dieser Zeit abgeschlossen wird, wird die Übertragung in einen Resynchronisationsstatus versetzt. Dadurch kann die nächste Übertragung, sofern vorhanden, ausgeführt werden.  In Szenarios mit einer hohen Auslastung des Quellen- oder Zielagenten ist es möglich, dass der Standardwert zu niedrig ist, sodass der Agent nicht schnell genug auf die Verhandlungsanfrage antworten kann. Dies gilt vor allem dann, wenn für einen Quellenagenten sehr viele Ressourcenmonitore definiert sind oder wenn seine Ressourcenmonitore Verzeichnisse überwachen, die sehr viele Dateien enthalten. Es kann aber auch auftreten, wenn sehr viele Übertragungsanforderungen an einen Agenten übergeben werden. In solchen Szenarios kann es erforderlich sein, den Wert dieser Eigenschaft auf 200.000 oder mehr zu erhöhen.	30 000
recoverableTransferRetryInterval	Die Wartezeit in Millisekunden zwischen der Erkennung eines behebbaren Übertragungsfehlers und dem Versuch, die Übertragung wiederaufzunehmen.	60000
senderTransferRetryInterval	Die Zeit in Millisekunden, nach der eine Übertragung wiederholt wird, die abgelehnt wurde, weil am Ziel bereits die maximale Anzahl an Übertragungen ausgeführt wird. Der Mindestwert ist 1000.	30 000
transferAckTimeout	Die Zeit in Millisekunden die eine Übertragung auf eine Bestätigung oder auf Daten der anderen Seite wartet, bevor ein erneuter Versuch stattfindet. Dies ist eine erweiterte Eigenschaft, deren Wert für die meisten Managed File Transfer-Konfigurationen nicht geändert werden muss.  Sobald ein vollständiges Datenfenster empfangen wird, werden vom empfangenden Agenten Bestätigungen an den sendenden Agenten gesendet. Bei Netzen mit einer Bandbreitenbeschränkung oder bei störanfälligen Netzen und hohen Einstellungen für <code>'agentWindowSize'</code> und <code>'agentChunkSize'</code> kann es vorkommen, dass der Standardwert nicht hoch genug ist. Dies kann zu einer unnötigen erneuten Übertragung von Daten zwischen den Agenten führen. Eine Erhöhung dieses Werts kann daher vorteilhaft sein und die Wahrscheinlichkeit verringern, dass eine Übertragung aufgrund eines langsamen Netzes in den Wiederherstellungsmodus wechselt.	60000
transferAckTimeoutRetries	Maximale Anzahl an Bestätigungswiederholungen für eine Übertragung ohne eine Antwort, bevor der Agent aufgibt und die Übertragung in den Zurückschreibungsmodus versetzt.	5

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
xmlConfigReloadInterval	<p>Das Intervall in Sekunden, das während der Laufzeit zwischen dem erneuten Laden der XML-Konfigurationsdateien durch den Agenten liegt. Wenn Sie verhindern möchten, dass der Agent XML-Konfigurationsdateien während der Laufzeit erneut lädt, müssen Sie diese Eigenschaft auf -1 setzen. Folgende XML-Konfigurationsdateien sind von dieser Eigenschaft betroffen:</p> <ul style="list-style-type: none"> <li>• ConnectDirectCredentials.xml</li> <li>• ConnectDirectNodeProperties.xml</li> <li>• ConnectDirectProcessDefinitions.xml</li> <li>• ProtocolBridgeCredentials.xml</li> <li>• ProtocolBridgeProperties.xml</li> <li>• UserSandboxes.xml</li> </ul>	30
<b>Eigenschaften für Traceerstellung und Protokollierung:</b>		
javaCoreTriggerFile	<p>Der vollständige Pfad zu einer Dateiposition, die der Agent überwacht. Wenn die Datei an der angegebenen Position vorhanden ist, löst der Agentenstart eine Javacore-Datei aus. Wenn Sie nach dem Agentenstart eine Datei an dieser Position aktualisieren, löst der Agent erneut eine Javacore-Datei aus.</p> <p>Ein separater Thread führt für diese Datei alle 30 Sekunden eine Abfrage durch, um zu prüfen, ob die Datei erstellt oder aktualisiert wurde. Wenn die Datei nach der letzten Abfrage erstellt oder aktualisiert wurde, generiert der Agent eine Java-Core-Dump-Datei in einem der folgenden Verzeichnisse:</p> <ul style="list-style-type: none"> <li>• UNIX: <code>MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name</code></li> <li>• Linux: <code>MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/agents/agent_name</code></li> <li>• Windows: <code>MQ_DATA_PATH\mqft\logs\coordination_qmgr_name\agents\agent_name</code></li> </ul> <p>Wenn Sie diese Eigenschaft angeben, gibt der Agent beim Start die folgende Nachricht aus:</p> <pre>BFGAG0092I The &lt;insert_0&gt; file will be used to request JVM diagnostic information.</pre> <p>Der Wert dieser Eigenschaft kann Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie unter „<a href="#">Verwendung von Umgebungsvariablen in MFT-Eigenschaften</a>“ auf Seite 179.</p>	--
trace	<p>Die beim Start des Agenten zu verwendende Tracespezifikation. Dies ist eine durch Kommas getrennte Liste mit Klassen und/oder Paketen, dem Gleichheitszeichen und einer Tracestufe.</p> <p>Wenn Sie beispielsweise die Klasse <code>com.ibm.wmqfte.agent.Agent</code> und die Klassen im Paket <code>com.ibm.wmqfte.commandhandler</code> ab dem Start des Agenten verfolgen möchten, fügen Sie der Datei <code>agent.properties</code> den folgenden Eintrag hinzu:</p> <pre>trace=com.ibm.wmqfte.agent.Agent,com.ibm.wmqfte.com mandhandler=all</pre> <p>Sie können auch mehrere Tracespezifikationen in einer durch Doppelpunkte getrennten Liste angeben. Beispiel:</p> <pre>tra ce=com.ibm.wmqfte.agent.Agent=all:com.ibm.wmqfte.com mandhandler=moderate</pre> <p>Die spezielle Tracespezifikation <code>=all</code> wird verwendet, um einen Trace für den Agenten und die Java Message Queuing Interface (JMQUI) zu erstellen, die die gesamte Kommunikation mit dem Agentenwarteschlangenmanager ausführen. Fügen Sie dazu den folgenden Eintrag zur Datei <code>agent.properties</code> hinzu:</p> <pre>trace==all</pre> <p>Sofern von Ihrem IBM Ansprechpartner nicht anders angegeben, verwenden Sie die Tracespezifikation <code>com.ibm.wmqfte=all</code> wie folgt:</p> <pre>trace=com.ibm.wmqfte=all</pre>	--

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)






Eigenschaftname	Beschreibung	Standardwert
outputLogFiles	Die Gesamtzahl der aufzubewahrenden Dateien des Typs output . log. Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	5
outputLogSize	Die maximale Größe in MB, die jede Datei output . log erreichen kann, bevor die Ausgabe in die nächste Datei geleitet wird. Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	1
outputLogEncoding	Die Zeichencodierung, die der Agent beim Schreiben von Daten in die Datei output . log verwendet.	Die Standardzeichencodierung der Plattform, auf welcher der Agent betrieben wird.
traceFiles	Die Gesamtanzahl der zu speichernden Tracedateien. Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	5
traceSize	Die maximale Größe (in MB) der einzelnen Tracedateien; sobald diese Größe erreicht ist, wird der Trace in eine Folgedatei geschrieben. Dieser Wert gilt für den Prozesscontroller eines Agenten und für den Agenten selbst.	20
traceMaxBytes	Die maximale Menge der Nachrichtendaten, die in die Tracedatei ausgegeben wird.	4096 Byte
logTransferRecovery	Wenn diese Eigenschaft auf den Wert 'true' gesetzt wird, werden Diagnoseereignisse im Ereignisprotokoll des Agenten in der Datei output0 . log gemeldet, sobald eine Übertragung in den Wiederherstellungsstatus versetzt wird.	Der Standardwert für IBM MQ 9.0.0.0 und höher ist 'true'.
logCapture	Erfasst Übertragungsanforderungsnachrichten, die an diesen Agenten übergeben werden, und Protokollnachrichten, die vom Agenten an den Koordinationswarteschlangenmanager veröffentlicht werden. Die erfassten Nachrichten können bei der Behebung von Übertragungsfehlern hilfreich sein. Erfasste Nachrichten werden in Dateien im Agentenprotokollverzeichnis capture? . log gespeichert. Das Symbol ? steht für einen numerischen Wert. Die Datei mit der Nummer 0enthält die neuesten erfassten Nachrichten.	false
logCaptureFileSize	Gibt die maximale Größe einer Aufzeichnungsdatei in Megabytes an.	10
logCaptureFiles	Gibt an, wie viele Aufzeichnungsdateien maximal aufbewahrt werden, bevor die älteste Datei gelöscht wird.	10
logCaptureFilter	Ein regulärer Java-Ausdruck, den der Agent zum Abgleichen des Themennamens der Nachricht verwendet. Es werden nur die dem regulären Ausdruck entsprechenden Nachrichten erfasst.	.* (alle abgleichen)
  resourceMonitor-Log	<p>Schlüssel/Wert-Paar für Ressourcenmonitor und Betrieb zum Aktivieren bzw. Inaktivieren der Protokollierung.</p> <p>Folgende Werte sind möglich:</p> <ul style="list-style-type: none"> <li>• info</li> <li>• moderate</li> <li>• verbose</li> <li>• off</li> </ul> <p>For example:</p> <ul style="list-style-type: none"> <li>• resourceMonitorLog=MON1,MON2=info:MON3=off</li> </ul> <p>Aktivieren Sie die Protokollierung für MON1 und MON2und inaktivieren Sie die Protokollierung für MON3.</p> <ul style="list-style-type: none"> <li>• resourceMonitorLog=info</li> </ul> <p>Aktivieren Sie die Protokollierung auf info-Ebene für alle Ressourcenmonitore.</p> <p>Die Ressourcenüberwachungsprotokolle werden in eine Datei mit dem Namen resmoneventN . loggeschrieben, wobei N für eine Zahl steht, z. B. resmonevent0 . log.</p> <p> <b>Achtung:</b> Alle Ressourcenmonitore eines Agenten schreiben in dieselbe Protokolldatei.</p> <p>Weitere Informationen finden Sie im Abschnitt <a href="#">Protokollierung der MFT-Ressourcenüberwachung</a>.</p>	info
  resourceMonitor-LogFileSize	Gibt die maximale Größe einer Aufzeichnungsdatei in Megabytes an.	20

Tabella 39. Erweiterte Agenteneigenschaften (Forts.)



Eigenschaftsname	Beschreibung	Standardwert
  resourceMonitor-LogFiles	Gibt an, wie viele Aufzeichnungsdateien maximal aufbewahrt werden, bevor die älteste Datei gelöscht wird.	10
<b>Eigenschaften der Übertragungsgrenzwerte:</b>		
maxDestinationTransfers	Die maximale Anzahl gleichzeitig ablaufender Übertragungen, die der Zielagent zu einem bestimmten Zeitpunkt verarbeitet. Jede an einen Agenten übergebene Übertragungsanforderung wird auf diese Gesamtzahl angerechnet, unabhängig von der Anzahl der Dateien, die zum Erfüllen der Anforderung übertragen werden. Das bedeutet, dass eine Übertragungsanforderung, in der eine einzelne Datei übertragen wird, ebenso gezählt wird wie eine Übertragungsanforderung, in der 10 Dateien übertragen werden.  Wenn der Zielagent das durch die Eigenschaft 'maxDestinationTransfers' angegebene Limit erreicht hat, stellt er Übertragungen in Warteschlangen.  Falls die Summe der Werte für die Agenteneigenschaften 'maxSourceTransfers', 'maxDestinationTransfers' und 'maxQueuedTransfers' den Wert der Einstellung MAXDEPTH der Statusspeicherwarteschlange (SYSTEM.FTE.STATE.Agentenname) überschreitet, wird der Agent nicht gestartet.	25 (bei allen Agenten außer Connect:Direct) 5 (bei Connect:Direct-Bridgeagenten)
maxFilesForTransfer	Die maximale Anzahl der Übertragungselemente, die für eine einzelne verwaltete Übertragung zulässig sind. Wenn eine Übertragung mehr Elemente enthält, als im Wert 'maxFilesForTransfer' angegeben sind, schlägt die verwaltete Übertragung fehl und es werden keine Übertragungselemente verarbeitet.  Durch das Festlegen dieser Eigenschaft wird verhindert, dass aufgrund einer fehlerhaften Übertragungsanforderung versehentlich zu viele Dateien übertragen werden, beispielsweise, wenn ein Benutzer versehentlich die Übertragung des Stammverzeichnisses / auf einem UNIX-System verlangt.	5000
maxSourceTransfers	Die maximale Anzahl gleichzeitig ablaufender Übertragungen, die der Quellenagent zu einem bestimmten Zeitpunkt verarbeitet. Jede an einen Agenten übergebene Übertragungsanforderung wird auf diese Gesamtzahl angerechnet, unabhängig von der Anzahl der Dateien, die zum Erfüllen der Anforderung übertragen werden. Das bedeutet, dass eine Übertragungsanforderung, in der eine einzelne Datei übertragen wird, ebenso gezählt wird wie eine Übertragungsanforderung, in der 10 Dateien übertragen werden.  Wenn der Quellenagent das durch die Eigenschaft 'maxSourceTransfers' angegebene Limit erreicht hat, stellt er Übertragungen in Warteschlangen.  Falls die Summe der Werte für die Agenteneigenschaften 'maxSourceTransfers', 'maxDestinationTransfers' und 'maxQueuedTransfers' den Wert der Einstellung MAXDEPTH der Statusspeicherwarteschlange (SYSTEM.FTE.STATE.Agentenname) überschreitet, wird der Agent nicht gestartet.	25 (bei allen Agenten außer Connect:Direct-Bridgeagenten) 5 (bei Connect:Direct-Bridgeagenten)
maxQueuedTransfers	Die maximale Anzahl anstehender Übertragungen, die von einem Quellenagent in die Warteschlange gestellt werden können, bis der Agent eine neue Übertragungsanforderung zurückweist. Sie können diese Eigenschaft so einstellen, dass trotz einer Überschreitung der Grenzwerte von 'maxDestinationTransfers' und 'maxSourceTransfers' neue Übertragungsanforderungen, die Sie jetzt stellen, akzeptiert, in eine Warteschlange eingereiht und dann ausgeführt werden.  Die Reihenfolge, in der die Übertragungsanforderungen aus der Warteschlange ausgeführt werden, wird durch deren Priorität und die Dauer bestimmt, die sie sich in der Warteschlange befinden. Ältere Anforderungen und Anforderungen mit hoher Priorität werden vorrangig verarbeitet. Übertragungen mit niedriger Priorität, die sich bereits lange Zeit in der Warteschlange befinden, werden vor neueren Übertragungen mit höherer Priorität ausgewählt.  Falls die Summe der Werte für die Agenteneigenschaften 'maxSourceTransfers', 'maxDestinationTransfers' und 'maxQueuedTransfers' den Wert der Einstellung MAXDEPTH der Statusspeicherwarteschlange (SYSTEM.FTE.STATE.Agentenname) überschreitet, wird der Agent nicht gestartet.	1000
  <b>Eigenschaften für das Zeitlimit für die Übertragungswiederherstellung:</b>		

Tabelle 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<b>V 9.1.0</b> <b>V 9.1.0</b> transferRecovery- Timeout	Geben Sie (in Sekunden) an, wie lange ein Quellenagent versuchen soll, eine blockierte Dateiübertragung wiederherzustellen.  Wird diese Eigenschaft nicht gesetzt, wird der Agent standardmäßig den Versuch so lange wiederholen, bis die Übertragung erfolgreich wiederhergestellt werden konnte. Sie können für die Eigenschaft zum Festlegen des Zeitlimits für die Übertragungswiederherstellung die folgenden Werte angeben:  <b>-1</b> Der Agent wiederholt den Versuch, die blockierte Übertragung wiederherzustellen, bis die Übertragung abgeschlossen wurde. Diese Option entspricht dem Standardverhalten des Agenten, wenn die Eigenschaft nicht gesetzt wird.  <b>0</b> Der Agent stoppt die Dateiübertragung, sobald die Wiederherstellung einsetzt.  <b>&gt;0</b> Der Agent wiederholt den Versuch, die blockierte Übertragung wiederherzustellen, bis der in Form eines positiven Integerwerts angegebene Zeitraum (in Sekunden) abgelaufen ist. Beispiel: <code>transferRecoveryTimeout=21600</code> gibt an, dass der Agent seit dem Eintritt in die Wiederherstellung 6 Stunden lang versucht, die Übertragung wiederherzustellen. Der Maximalwert für diesen Parameter ist 999999999.	-1
<b>Eigenschaften für die Benutzerexitroutine:</b>		
agentForceConsistentPathDelimiters	Als Pfadtrennzeichen wird in den Informationen zur Quellen- und Zieldatei, die in den Übertragungsexits bereitgestellt werden, der UNIX-Schrägstrich (/) verwendet. Gültige Optionen sind <code>true</code> und <code>false</code> .	false
destinationTransferEndExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine zum Zielübertragungsende implementieren.	Kein Standardwert
destinationTransferStartExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine zum Zielübertragungsstart implementieren.	Kein Standardwert
exitClassPath	Gibt eine plattformspezifische, von Zeichen begrenzte Liste von Verzeichnissen an, die als Klassenpfad für Benutzerexitroutinen agieren.  Vor Einträgen in diesem Klassenpfad wird das Exitverzeichnis des Agenten durchsucht.	Verzeichnis <code>exits</code> des Agenten
exitNativeLibraryPath	Gibt eine plattformspezifische, von Zeichen begrenzte Liste von Verzeichnissen an, die als Pfad der nativen Bibliothek für Benutzerexitroutinen agieren.	Verzeichnis <code>exits</code> des Agenten
ioMaxRecordLength	Die maximale Satzlänge in Byte, die für eine satzorientierte Datei unterstützt werden kann. Managed File Transfer kann das Schreiben in satzorientierte Dateien mit beliebiger Satzlänge unterstützen. Da lange Datensätze jedoch zu Fehlern aufgrund abnormaler Speicherbedingungen führen können, wird die maximale Satzlänge zur Vermeidung derartiger Fehler standardmäßig auf 64 K beschränkt. Beim Auslesen satzorientierter Dateien muss ein ganzer Datensatz in einen einzelnen Übertragungsblock passen. Daher ist die Satzlänge zusätzlich durch die Größe des Übertragungsblocks beschränkt. Diese Eigenschaft wird nur für satzorientierte Dateien des Ein-/Ausgabebenutzerexits verwendet.	64 KB
monitorExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Übertragungsexitroutine implementieren. Weitere Informationen finden Sie im Abschnitt <a href="#">Benutzerexits für MFT-Ressourcenüberwachungen</a> .	Kein Standardwert
protocolBridgeCredentialExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine für Protokollbrideglauberechtigungs-nachweise implementieren. Weitere Informationen finden Sie im Abschnitt <a href="#">Berechtigungs-nachweise für einen Dateiserver mittels Exitklassen zuordnen</a> .	Kein Standardwert.
sourceTransferEndExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Exitroutine zum Quellenübertragungsende implementieren.	Kein Standardwert
sourceTransferStartExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Exitroutine zum Quellenübertragungsstart implementieren.	Kein Standardwert
IOExitClasses	Gibt eine durch Kommas getrennte Liste von Klassen an, die eine Benutzerexitroutine für Ein-/Ausgaben implementieren. Listen Sie nur die Klassen auf, die die Schnittstelle 'IOExit' implementieren; listen Sie also keine Klassen auf, die die anderen Benutzerexits für Ein-/Ausgaben implementieren (beispielsweise <code>IOExitResourcePath</code> und <code>IOExitChannel</code> ). Weitere Informationen finden Sie im Abschnitt <a href="#">MFT-Übertragungs-E/A-Benutzerexits verwenden</a> .	Kein Standardwert.
<b>IBM MQ-Clientkomprimierung:</b>		



Tabella 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
agentDataCompression	<p>Diese Eigenschaft wird nur für Clientverbindungen unterstützt.</p> <p>Eine durch Kommas getrennte Liste mit Komprimierungstypen für die Übertragung von Dateidaten, über die mit dem fernen IBM MQ-Server verhandelt wird. Informationen zu diesen Komprimierungstypen finden Sie im folgenden Abschnitt: <a href="#">Message data compression list</a> (Liste zur Nachrichtendatenkomprimierung).</p> <p>Die Werte werden auf ihre Gültigkeit überprüft und anschließend in der angezeigten Reihenfolge als Eigenschaften an den Clientkanal des Agenten übergeben. Der IBM MQ-Client verarbeitet nun Vereinbarungen zwischen diesem Clientkanal und dem fernen Serverkanal, um den kleinsten gemeinsamen Nenner der Komprimierungseigenschaften auf den beiden Kanälen zu finden. Wenn es keine Übereinstimmung gibt, wird immer MQCOMPRESS_NONE ausgewählt.</p>	MQCOMPRESS_NONE
agentHeaderCompression	<p>Diese Eigenschaft wird nur für Clientverbindungen unterstützt.</p> <p>Eine durch Kommas getrennte Liste mit Komprimierungstypen für die Übertragung von Headerdaten, über die mit dem fernen IBM MQ-Server verhandelt wird. Die gültigen Werte lauten MQCOMPRESS_NONE bzw. MQCOMPRESS_SYSTEM. Informationen zu diesen Komprimierungstypen finden Sie im folgenden Abschnitt: <a href="#">HdrCompList [2] (MQLONG)</a>.</p> <p>Die Werte werden auf ihre Gültigkeit überprüft und anschließend in der angezeigten Reihenfolge als Eigenschaften an den Clientkanal des Agenten übergeben. Der IBM MQ-Client verarbeitet nun Vereinbarungen zwischen diesem Clientkanal und dem fernen Serverkanal, um den kleinsten gemeinsamen Nenner der Komprimierungseigenschaften auf den beiden Kanälen zu finden. Wenn es keine Übereinstimmung gibt, wird immer MQCOMPRESS_NONE ausgewählt.</p>	MQCOMPRESS_NONE
  <b>z/OS-spezifische Eigenschaften:</b>		
  adminGroup	<p>Eine Sicherheitsmanagergruppe. Mitglieder dieser Gruppe können:</p> <ul style="list-style-type: none"> <li>den Agenten mit dem Befehl <b>fteStartAgent</b> starten,</li> <li>den Agenten mit dem Befehl <b>fteStopAgent</b> stoppen,</li> <li>mit dem Befehl <b>fteSetAgentTraceLevel</b> einen Trace für den Agenten aktivieren oder inaktivieren und</li> <li>Aktivieren oder inaktivieren Sie Protokolle für den Agenten mit dem Befehl <b>fteSetAgentLogLevel</b>.</li> <li>Zeigen Sie Einzelheiten zu einem lokalen Agenten an, indem Sie den Befehl <b>fteShowAgentDetails</b> mit dem Parameter <b>-d</b> ausführen.</li> </ul> <p>Definieren Sie eine Sicherheitsmanagergruppe, z. B. <i>MFTADMIN</i>, und fügen Sie dann die Benutzer-ID für gestartete Tasks und Administrator-TSO-IDs zu dieser Gruppe hinzu. Bearbeiten Sie die Agenteneigenschaftendatei und setzen Sie die Eigenschaft <b>adminGroup</b> auf den Namen dieser Sicherheitsmanagergruppe.</p> <pre>adminGroup=MFTADMIN</pre>	--
  bpxwdynAllocAdditionalOptions	<p>Managed File Transfer verwendet zum Erstellen und Öffnen von z/OS-Datensätzen die Textschnittstelle BPXWDYN. Wenn BPXWDYN standardmäßig für die Datasetzuordnung verwendet wird, stellt Managed File Transfer, sofern möglich, sicher, dass das Datengerät gemountet ist (dies ist für Datensätze auf Festplatten nicht nötig, für Banddatensätze hingegen schon). Da die BPXWDYN-Optionen in einigen Umgebungen nicht unterstützt werden, können Sie dieses Verhalten mit dieser Eigenschaft ändern. Bei der Übertragung an ein Dataset können Optionen für BPXWDYN auch in der Befehlszeile angegeben werden. Diese Optionen kommen zu den über diese Eigenschaft definierten Optionen hinzu.</p> <p>Bei Verwendung der Eigenschaft <b>bpxwdynAllocAdditionalOptions</b> in der Datei <code>agent.properties</code> dürfen einige BPXWDYN-Optionen nicht angegeben werden. Eine Liste dieser Eigenschaften finden Sie im Abschnitt <b>BPXWDYN-Eigenschaften</b>, die nicht mit MFT verwendet werden dürfen.</p>	<p>Die Standardwerte lauten wie folgt:</p> <ul style="list-style-type: none"> <li>MOUNT für z/OS V1R8 und höher</li> </ul>
  armELEMTYPE	<p>Optionale Eigenschaft. Wenn der Agent so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMTYPE, der in der zugehörigen ARM-Richtlinie festgelegt ist. Für einen Agent muss ELEMTYPE auf SYSBFGAG gesetzt werden.</p>	Nicht festgelegt

Table 39. Erweiterte Agenteneigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<p><b>z/OS</b></p> <p><b>z/OS</b> armELEMENT</p>	Optionale Eigenschaft. Wenn der Agent so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMENT, der in der zugehörigen ARM-Richtlinie festgelegt ist. Der Wert von ELEMENT kann dem Namen des Agenten entsprechen.	Nicht festgelegt
<b>Weitere Eigenschaften:</b>		
<p><b>z/OS</b></p> <p><b>z/OS</b> legacyXMLMessageMQMDFormat</p>	<p>Vom Managed File Transfer-Agenten generierte XML-Nachrichten (z. B. Protokoll- und Übertragungsstatusnachrichten) werden nun an eine Warteschlange mit einem leeren MQMD-Formatfeld gesendet. In früheren Produktversionen war das MQMD-Formatfeld auf MQSTR gesetzt (eine Textnachrichten-Zeichenfolge). Wenn diese Eigenschaft auf "true" gesetzt ist, werden die vom Managed File Transfer-Agenten generierten XML-Nachrichten an eine Warteschlange gesendet, deren MQMD-Formatfeld auf MQSTR gesetzt ist.</p> <p><b>Anmerkung:</b> Das Nachrichtenformat der Antwortnachrichten des Agenten auf Befehle entspricht dem Format der Befehlsanforderung.</p> <p>Wenn das MQMD-Formatfeld auf MQSTR gesetzt ist und das MQ-Netz Kanäle mit Datenkonvertierung enthält, besteht die Gefahr, dass die XML-Nachrichten für Managed File Transfer-Befehle beschädigt werden.</p>	false
<p><b>V 9.1.0.5</b> adjustScheduleTimeForDaylightSaving</p>	<p>Wenn Ihr Unternehmen jeden Tag planmäßige Übertragungen durchführt, d. h. mit:</p> <ul style="list-style-type: none"> <li>• Parameter <b>-oia</b> auf Tage gesetzt und</li> <li>• Parameter <b>-tba</b> auf Quelle gesetzt</li> </ul> <p>im Befehl <code>fteCreateTransfer</code> eingestellt ist, setzen Sie dieses Attribut auf <code>true</code>, um die geplante Übertragungszeit um eine Stunde nach vorne zu verschieben, wenn die Uhren um eine Stunde vorgestellt werden, und um eine Stunde zurück, wenn die Uhren um eine Stunde zurückgestellt werden.</p> <p>Wenn Ihre geplante Überweisung beispielsweise um 1:00 Uhr nachts erfolgen soll, wird die Überweisung bei der Zeitumstellung um 2:00 Uhr nachts ausgeführt, und wenn die Uhren zurückgestellt werden, wird die Überweisung auf 1:00 Uhr zurückgesetzt.</p>	true

## Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

**V 9.1.0** [Zeitlimitoption für die Wiederherstellung von Dateiübertragungen](#)

[MFT-Sandboxes](#)

## Zugehörige Verweise

[„Java-Systemeigenschaften für MFT“ auf Seite 232](#)

Einige Managed File Transfer-Befehls- und Agenteneigenschaften müssen als Java-Systemeigenschaften definiert werden, da sie die Konfiguration für ältere Funktionen bereitstellen, die die Mechanismen der Befehle und Agenteneigenschaften nicht unterstützen.

[„SSL-Eigenschaften für MFT“ auf Seite 233](#)

Einige MFT-Eigenschaftendateien enthalten SSL-Eigenschaften. Mit SSL oder TLS können Sie in IBM MQ und Managed File Transfer unberechtigte Verbindungen zwischen Agenten und Warteschlangenmanagern verhindern und die Nachrichtenübertragungen zwischen Agenten und Warteschlangenmanagern verschlüsseln.

[fteCreateAgent](#)

[fteCreateBridgeAgent](#)

[fteCreateCDAgent](#)

[„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 179](#)

Ab IBM WebSphere MQ 7.5 können in Managed File Transfer-Eigenschaften Umgebungsvariablen für Datei- oder Verzeichnispositionen verwendet werden. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt).

## MFT-Agenten für mehrere Kanäle in einem Cluster konfigurieren

Soll in einer Clusterkonfiguration die IBM MQ-Unterstützung für mehrere Kanäle verwendet werden, müssen Sie zunächst die Eigenschaft 'agentMultipleChannelsEnabled' auf `true` setzen und anschließend die in diesem Abschnitt beschriebenen Schritte ausführen.

### Informationen zu diesem Vorgang

In einem Cluster wird die Unterstützung für mehrere Kanäle nur über die IBM MQ-Definitionen im Warteschlangenmanager des Zielagenten aktiviert.

Zusätzlich zu den Schritten der IBM MQ-Standardkonfiguration für einen Managed File Transfer-Agenten (siehe [MFT für erstmalige Verwendung konfigurieren](#)) müssen Sie die hier beschriebenen Schritte ausführen.

Die nachfolgenden Konfigurationsbeispiele veranschaulichen die Verwendung der `runmqsc`-Befehle.

### Vorgehensweise

1. Definieren Sie für jeden Kanal, der verwendet werden soll, einen Clusterempfängerkanal. Beispiel bei Verwendung von zwei Kanälen:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(CLUSRCVR) CLUSTER(MFTCLUSTER)
```

Dabei gilt:

- `ZIEL_WS_MANAGER` steht für den Namen des Warteschlangenmanagers des Zielagenten.
- `MFTCLUSTER` steht für den Namen des IBM MQ-Clusters.

Es wird empfohlen, die Namenskonvention `MFTCLUSTER.ZIEL_WS_MANAGER_n` für Kanäle zu verwenden; diese Konvention ist jedoch nicht verbindlich.

2. Definieren Sie für jeden Kanal einen Warteschlangenmanager-Aliasnamen. For example:

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) CLUSTER(MFTCLUSTER)
```

Sie müssen das `SYSTEM.FTE.DESTQMGRNAME_n` Namenskonvention für WS-Manager-Aliasnamen, da der sendende Agent nach WS-Manager-Aliasnamen dieses Formats sucht. Für `n` müssen jeweils fortlaufende Nummern angegeben werden, die bei 1 beginnen. Die Definitionen müssen clusterweit erfolgen, damit sie auf dem Warteschlangenmanager des Quellenagenten verfügbar sind.

Damit Quellen- und Zielagent die Nummer der Warteschlangenmanager-Aliasnamen korrekt ermitteln kann, darf für den Warteschlangenmanager **keine** Standardübertragungswarteschlange (XMITQ) definiert werden.

### Zugehörige Tasks

[MFT für erstmalige Verwendung konfigurieren](#)

### Zugehörige Verweise

„Die MFT agent.properties-Datei“ auf Seite 196

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens `agent.propertiesagent.properties`, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei `agent.properties` angegeben sein.

## MFT-Agenten für mehrere Kanäle konfigurieren: clusterunabhängig

Soll die IBM MQ-Unterstützung für mehrere Kanäle in einer Konfiguration aktiviert werden, bei der es sich nicht um einen Cluster handelt, müssen Sie zunächst die Eigenschaft 'agentMultipleChannelsEnabled' auf `true` setzen und anschließend die in diesem Abschnitt beschriebenen Schritte ausführen.

## Informationen zu diesem Vorgang

In einer Konfiguration, bei der es sich nicht um einen Cluster handelt, wird die Unterstützung mehrerer Kanäle über die IBM MQ-Definitionen im Warteschlangenmanager auf dem Quellen- und auf dem Zielagenten aktiviert.

Zusätzlich zu den Schritten der IBM MQ-Standardkonfiguration für einen Managed File Transfer-Agenten (siehe [MFT für erstmalige Verwendung konfigurieren](#)) müssen Sie die hier beschriebenen Schritte ausführen.

Bei den folgenden Schritten wird davon ausgegangen, dass die Kommunikation zwischen den Quellen- und Zielwarteschlangenmanagern über Sender-/Empfängerkanäle erfolgt.

Die nachfolgenden Konfigurationsbeispiele veranschaulichen die Verwendung der **runmqsc**-Befehle.

## Vorgehensweise

1. Definieren Sie im Warteschlangenmanager des Zielagenten für jeden Kanal, der verwendet werden soll, einen Empfängerkanal. Beispiel bei Verwendung von zwei Kanälen:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

Dabei ist DESTQMGRNAME der Name des Warteschlangenmanagers des Zielagenten.

Für Kanalnamen empfiehlt sich die Verwendung des Formats TO.DESTMGRNAME\_n, seine Verwendung ist jedoch nicht zwingend. Die Namen der Empfängerkanäle müssen den jeweiligen Senderkanälen des Warteschlangenmanagers des Quellenagenten entsprechen.

2. Definieren Sie im Warteschlangenmanager des Quellenagenten für jeden Kanal, der verwendet werden soll, eine Übertragungswarteschlange. Beispiel bei Verwendung von zwei Kanälen:

```
DEFINE QLOCAL(DESTQMGRNAME_1) USAGE(XMITQ)
DEFINE QLOCAL(DESTQMGRNAME_2) USAGE(XMITQ)
```

Für Übertragungswarteschlangen empfiehlt sich die Verwendung des Formats DESTMGRNAME\_n, seine Verwendung ist jedoch nicht zwingend. Auf die von Ihnen definierten Übertragungswarteschlangen wird in den folgenden Schritten von den Senderkanaldefinitionen und den Warteschlangenmanager-Aliasnamensdefinitionen verwiesen.

3. Definieren Sie im Warteschlangenmanager des Quellenagenten für jeden Kanal, der verwendet werden soll, einen Senderkanal. Beispiel bei Verwendung von zwei Kanälen:

```
DEFINE CHANNEL(TO.DESTQMGRNAME_1) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_1)
DEFINE CHANNEL(TO.DESTQMGRNAME_2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(DESTHOST:port)
XMITQ(DESTQMGRNAME_2)
```

Für Kanäle empfiehlt sich die Verwendung des Formats TO.DESTMGRNAME\_n, seine Verwendung ist jedoch nicht zwingend. Die Namen der Senderkanäle müssen den jeweiligen Empfängerkanälen im Warteschlangenmanager des Zielagenten entsprechen.

4. Definieren Sie im Warteschlangenmanager des Quellenagenten für jeden Kanal einen entsprechenden Warteschlangenmanager-Aliasnamen. Beispiel:

```
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_1) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_1)
DEFINE QREMOTE(SYSTEM.FTE.DESTQMGRNAME_2) RQMNAME(DESTQMGRNAME) XMITQ(DESTQMGRNAME_2)
```

Für Warteschlangenmanager-Aliasnamen muss SYSTEM.FTE.DESTQMGRNAME\_n als Namensformat verwendet werden, da der sendende Agent nach Warteschlangenmanager-Aliasnamen dieses Formats sucht. Für n müssen jeweils fortlaufende Nummern angegeben werden, die bei 1 beginnen.

Damit der Agent die Nummer der Warteschlangenmanager-Aliasnamen korrekt ermitteln kann, darf für den Warteschlangenmanager **keine** Standardübertragungswarteschlange (XMITQ) definiert werden.

## Zugehörige Tasks

MFT für erstmalige Verwendung konfigurieren

## Zugehörige Verweise

„Die MFT agent.properties-Datei“ auf Seite 196

Für jeden Managed File Transfer Agent gibt es eine eigene Eigenschaftendatei namens `agent.properties`, in der Informationen für die Verbindung des Agenten zum Warteschlangenmanager enthalten sein müssen. Auch Eigenschaften, die das Verhalten des Agenten ändern, können in der Datei `agent.properties` angegeben sein.

## enableUserMetadataOptions: Unterstützte benutzerdefinierte MFT-Metadaten-schlüssel

Wenn die Agenteneigenschaft **enableUserMetadataOptions** auf den Wert `true` gesetzt wird, werden bei der Angabe einer neuen Übertragungsanforderung die folgenden benutzerdefinierten Metadaten-schlüssel unterstützt.

*Tabelle 40. Metadaten-schlüssel*

Schlüsselname	Beschreibung	Standardwert
<code>com.ibm.wmqfte.insertRecordLineSeparator</code>	Für Textübertragungen. Wird dieser Schlüssel auf 'true' gesetzt, werden beim Lesen satzorientierter Dateien, z. B. z/OS-Datasets, Zeilentrennzeichen zwischen Datensätzen eingefügt.  Wird er auf 'false' gesetzt, werden beim Lesen satzorientierter Dateien keine Zeilentrennzeichen zwischen Datensätzen eingefügt.	true
<code>com.ibm.wmqfte.newRecordOnLineSeparator</code>	Für Textübertragungen. Wird dieser Schlüssel auf 'true' gesetzt, markieren Zeilentrennzeichen beim Schreiben in satzorientierte Dateien, z. B. z/OS-Datasets, den Anfang eines neuen Datensatzes und werden selbst nicht als Teil der Daten geschrieben.  Wird er auf 'false' gesetzt, werden Zeilentrennzeichen beim Schreiben in satzorientierte Dateien wie alle anderen Zeichen behandelt (d. h., nicht als Trennzeichen zwischen Datensätzen).	true
<code>com.ibm.wmqfte.convertLineSeparators</code>	Für Textübertragungen. Gibt an, ob die Zeilentrennzeichenfolgen CRLF und LF in Zeilentrennzeichenfolgen für das Ziel umgewandelt werden. Diese Umwandlung findet zurzeit nur in folgenden Fällen statt: <ol style="list-style-type: none"><li>Der benutzerdefinierte Metadaten-schlüssel <code>com.ibm.wmqfte.newRecordOnLineSeparator</code> wird auf 'false' gesetzt und die Übertragung erfolgt in eine satzorientierte Datei.</li><li>Der benutzerdefinierte Metadaten-schlüssel <code>com.ibm.wmqfte.insertRecordLineSeparator</code> wird auf 'false' gesetzt und die Übertragung erfolgt aus einer satzorientierten Datei.</li></ol>	true

## Zugehörige Verweise

`fteCreateTransfer-Parameter -md`

## Zugehörige Informationen

„Tabelle 39“ auf Seite 198

## Die MFT-Datei 'logger.properties'

Für die Managed File Transfer-Protokollfunktion sind eine Reihe von Konfigurationseigenschaften vorhanden. Diese Eigenschaften werden in der Datei `logger.properties` definiert, die sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` befindet.

Ab IBM WebSphere MQ 7.5 können Sie Umgebungsvariablen in einigen Managed File Transfer-Eigenschaften verwenden, die Datei- oder Verzeichnispositionen darstellen. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B.

an den Benutzer, der den Prozess ausführt). Weitere Informationen finden Sie unter „[Verwendung von Umgebungsvariablen in MFT-Eigenschaften](#)“ auf Seite 179.

**Anmerkung:** **Windows** Bei der Angabe von Dateipfaden unter Windows muss der als Trennzeichen verwendete umgekehrte Schrägstrich (\) doppelt (\\) angegeben werden (d. h. mit einem \ als Escapezeichen). Alternativ können Sie einen einfachen Schrägstrich (/) als Trennzeichen verwenden. Weitere Informationen zur Verwendung von Escapezeichen in Java-Eigenschaftendateien unter Oracle finden Sie in [Javadoc for the Properties class](#).

## Eigenschaften von Verbindungen im Bindungsmodus

Tabelle 41. Verbindungseigenschaften für den Bindungsmodus		
Eigenschaftsname	Beschreibung	Standardwert
wmqfte.logger.type	Der verwendete Dateimodus. Setzen Sie diesen Wert auf FILE oder DATABASE.	Kein Standardwert
wmqfte.max.transaction.messages	Die maximale Anzahl Nachrichten, die in einer Transaktion verarbeitet werden, bevor die Transaktion festgeschrieben wird. Bei Verwendung der Umlaufprotokollierung steht einem Warteschlangenmanager ein fester Speicherbereich für unvollständige Daten zur Verfügung. Stellen Sie sicher, dass diese Eigenschaft einen möglichst niedrigen Wert erhält, damit der verfügbare Speicherplatz ausreicht.	50
wmqfte.max.transaction.time	Die maximale Zeit in Millisekunden zwischen Transaktionsfestschreibungen.	5000
wmqfte.max.consecutive.reject	Die maximale Anzahl Nachrichten, die nacheinander zurückgewiesen werden kann (d. h., ohne dass eine gültige Nachricht erkannt wird).  Bei Überschreiten dieses Wertes geht die Protokollfunktion davon aus, dass der Fehler nicht bei den Nachrichten selbst, sondern bei der Konfiguration liegt. Wenn Sie beispielsweise die Spalte für den Agentennamen in der Datenbank so schmal machen, dass die Agentennamen dafür zu lang sind, werden alle Nachrichten, die auf Agenten verweisen, zurückgewiesen.	50
wmqfte.reject.queue.name	Der Name einer Warteschlange, in die die Protokollfunktion Nachrichten einreicht, die sie nicht verarbeiten kann. Bei Verwendung einer Datenbankprotokollfunktion finden Sie im Abschnitt <a href="#">Fehlerbehandlung und Zurückweisung in der MFT-Protokollfunktion</a> Hinweise zu den Nachrichten, die in diese Warteschlange eingereicht werden können.	SYSTEM.FTE.LOG.RJCT.Name_der_Protokollfunktion
wmqfte.command.queue.name	Der Name einer Warteschlange, aus der die Protokollfunktion Befehlsnachrichten abrufen, über die ihr Verhalten gesteuert wird.	SYSTEM.FTE.LOG.CMD.Name_der_Protokollfunktion
wmqfte.queue.manager	Der Warteschlangenmanager, mit dem sich die Protokollfunktion verbindet. Dieser Parameter ist erforderlich und die einzige benötigte Angabe für Verbindungen im Bindungsmodus zum Warteschlangenmanager. (Die Eigenschaften für Verbindungen mit einem fernen Warteschlangenmanager finden Sie im Abschnitt <a href="#">Tabelle 42 auf Seite 230</a> .)	Kein Standardwert

Tabelle 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)




Eigenschaftsname	Beschreibung	Standardwert
wmqfte.message.source.type	<p>Einer der folgenden Werte:</p> <p><b>automatic subscription</b> Dies ist der Standardwert. Die Protokollfunktion erstellt und verwendet in dem in 'SYSTEM.FTE/Log/#' definierten Warteschlangenmanager eine eigene permanente, verwaltete Subskription. Dieser Wert ist für die meisten Szenarios geeignet.</p> <p><b>administrative subscription</b> Wenn die automatische Subskription nicht geeignet ist, können Sie eine andere Subskription definieren (beispielsweise mithilfe von IBM MQ Explorer, MQSC oder PCF) und die Protokollfunktion anweisen, diese Subskription zu verwenden. Mit dem folgenden Wert beispielsweise können Sie den Protokollbereich so aufteilen, dass von einer Protokollfunktion Agenten von A-H, von einer zweiten Protokollfunktion die von I-P und von einer dritten Protokollfunktion die von Q-Z verarbeitet werden.</p> <p><b>queue</b> Wenn es aufgrund der IBM MQ-Topologie zu umständlich ist, eine Subskription für die Protokollfunktion zu erstellen, können Sie stattdessen eine Warteschlange verwenden. Konfigurieren Sie IBM MQ so, dass die Warteschlange die Nachrichten empfängt, die normalerweise von einer Subskription für SYSTEM.FTE/Log/# auf dem Koordinationswarteschlangenmanager.</p>	automatic subscription
wmqfte.message.source.name	Wenn der Nachrichtenquellentyp administrative subscription oder queue lautet, ist dies der Name der zu verwendenden Subskription oder Warteschlange. Diese Eigenschaft wird ignoriert, wenn der Quellentyp automatic subscription lautet.	Kein Standardwert
wmqfte.database.credentials.file	<p>Der Name der Datei, die den Benutzernamen und das Kennwort zum Herstellen einer Verbindung zur Datenbank enthält.</p> <p>Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.</p> <p>Weitere Informationen finden Sie im Abschnitt <a href="#">MFT-Dateiformat für Berechtigungsnachweise</a>.</p>	<p> Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter <a href="#">MQMFTCredentials.xml</a> unter z/OS konfigurieren.</p> <p> Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt <a href="#">MQMFTCredentials.xml</a> konfigurieren.</p> <p> Weitere Informationen zum Erstellen der Authentifizierungsbe-rechtigungsdatei finden Sie unter <a href="#">MFT- und IBM MQ-Authentifizierungsnachweis</a>.</p>

Tabelle 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<p>wmqfte.database.driver</p>	<p>Die Position der JDBC-Treiberklassen für die Datenbank. Dies ist normalerweise der Pfad und Dateiname einer JAR-Datei.</p> <p><b>AIX</b> Für den Db2-Treiber des Typs 2 auf AIX-Systemen ist beispielsweise die Datei /opt/IBM/db2/V9.5/java/db2jcc.jar erforderlich.</p> <p><b>Windows</b> Auf Windows-Systemen wird als Pfadtrennzeichen der Schrägstrich (/) verwendet (Beispiel: C:/Program Files/IBM/SQLLIB/java/db2jcc.jar).</p> <p><b>z/OS</b> Geben Sie unter z/OS den vollständigen Pfad der Datei db2jcc.jar an. Beispiel: wmqfte.database.driver=/db2/db2v10/jdbc/classes/db2jcc.jar.</p> <p><b>z/OS</b> Auf z/OS-Systemen müssen alle folgenden JAR-Dateien referenziert werden:</p> <ul style="list-style-type: none"> <li>• db2jcc.jar</li> <li>• db2jcc_license_cisuz.jar</li> <li>• db2jcc_javax.jar</li> </ul> <p>Wenn Ihr Datenbanktreiber aus mehreren JAR-Dateien besteht (für Db2 V9.1 sind beispielsweise eine JAR-Treiberdatei und eine JAR-Lizenzdatei erforderlich), schließen Sie alle diese JAR-Dateien in diese Eigenschaft ein. Bei Angabe mehrerer Dateinamen müssen diese mit dem auf Ihrer Plattform üblichen Klassenpfadtrennzeichen getrennt werden; auf Windows-Systemen ist dies das Semikolon (;), auf allen anderen Plattformen der Doppelpunkt (:).</p>	<p>Kein Standardwert</p>
<p>wmqfte.database.exclude.duplicate.metadata</p>	<p>Steuert, ob Einträge in der Metadaten-tabelle gespeichert werden. Diese Tabelle enthält übergeordnete Informationen anderer Tabellen des Schemas der Datenbankprotokollfunktion. Setzen Sie diesen Wert auf true oder false. Die Metadateneinträge werden standardmäßig nicht mehr in dieser Tabelle gespeichert, da dies eine Art der Verschwendung von Datenbankspeicher durch Datenduplizierung ist. Die folgenden Eigenschafteneinträge in den jeweils angegebenen Tabellen enthalten dieselben Daten:</p> <ul style="list-style-type: none"> <li>• com.ibm.wmqfte.SourceAgent: TRANSFER_EVENT oder CALL_REQUEST</li> <li>• com.ibm.wmqfte.DestinationAgent: TRANSFER_EVENT</li> <li>• com.ibm.wmqfte.MqmdUser: TRANSFER_EVENT oder CALL_REQUEST</li> <li>• com.ibm.wmqfte.OriginatingUser: TRANSFER_EVENT oder CALL_REQUEST</li> <li>• com.ibm.wmqfte.OriginatingHost: TRANSFER_EVENT oder CALL_REQUEST</li> <li>• com.ibm.wmqfte.TransferId: TRANSFER oder CALL_REQUEST</li> <li>• com.ibm.wmqfte.JobName: TRANSFER oder CALL_REQUEST</li> </ul> <p>Wenn der Wert dieser Eigenschaft auf false gesetzt ist, werden die Metadateneinträge in der Metadaten-tabelle gespeichert.</p>	<p>true</p>



Tabelle 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.database.host	<p>Nur für Db2:</p> <p>Ab IBM WebSphere MQ 7.5 der Hostname des Datenbankservers, zu dem über einen Typ-4-JDBC-Treiber eine Verbindung hergestellt werden soll. Wenn ein Wert für diese Eigenschaft angegeben wird, muss auch ein Wert für <code>wmqfte.database.port</code> angegeben werden. Wenn keine der beiden Eigenschaften definiert ist, stellt die Datenbankprotokollfunktion eine Verbindung über den standardmäßigen Typ-2-JDBC-Treiber her.</p> <p>Wenn für diese Eigenschaft ein Wert angegeben ist, muss für diese Protokollfunktion eine Datei mit Berechtigungsnachweisen vorhanden und zugänglich sein (d. h. ein mit <code>wmqfte.database.credentials.file</code> definierter Dateipfad). Diese Datei bestimmt den Benutzernamen und das Kennwort für die Verbindung mit der Datenbank, selbst wenn sich die Datenbank auf dem lokalen System befindet.</p>	Kein Standardwert
wmqfte.database.name	Der Name der Datenbankinstanz (oder des Subsystems bei der Verwendung von Db2 for z/OS), die die Managed File Transfer-Protokolltabellen enthält.	Kein Standardwert
wmqfte.database.type	Das verwendete Datenbankmanagementsystem Db2 oder Oracle). Setzen Sie diesen Wert auf <code>db2</code> oder <code>oracle</code> .	db2
wmqfte.database.port	<p>Nur für Db2:</p> <p>Ab IBM WebSphere MQ 7.5 die Portnummer des Datenbankservers, zu dem über einen Typ-4-JDBC-Treiber eine Verbindung hergestellt werden soll. Wenn ein Wert für diese Eigenschaft angegeben wird, muss auch ein Wert für <code>wmqfte.database.host</code> angegeben werden. Wenn keine der beiden Eigenschaften definiert ist, stellt die Datenbankprotokollfunktion eine Verbindung über den standardmäßigen Typ-2-JDBC-Treiber her.</p> <p>Wenn für diese Eigenschaft ein Wert angegeben ist, muss für diese Protokollfunktion eine Datei mit Berechtigungsnachweisen vorhanden und zugänglich sein (d. h. ein mit <code>wmqfte.database.credentials.file</code> definierter Dateipfad). Diese Datei bestimmt den Benutzernamen und das Kennwort für die Verbindung mit der Datenbank, selbst wenn sich die Datenbank auf dem lokalen System befindet.</p>	Kein Standardwert
wmqfte.database.schema	<p>Nur für Db2:</p> <p>Das Datenbankschema, das die Managed File Transfer-Protokolltabellen enthält. In den meisten Fällen ist der Standardwert geeignet, aber Sie können bei Bedarf abhängig von standortspezifischen Datenbankspekten einen alternativen Wert angeben.</p>	FTELOG

Tabella 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.database.native.library.path	<p>Der Pfad, in dem sich die nativen Bibliotheken befinden, die der von Ihnen ausgewählte Datenbanktreiber (falls vorhanden) benötigt.</p> <p><b>AIX</b> Für den Db2-Treiber des Typs 2 auf AIX-Systemen sind beispielsweise Bibliotheken aus /opt/IBM/db2/V9.5/lib32/ erforderlich. Alternativ zu dieser Eigenschaft können Sie mit anderen Methoden die Systemeigenschaft 'java.library.path' festlegen.</p> <p><b>Solaris</b> Unter Solaris müssen Sie vor der Ausführung des Befehls <b>fteStartLogger</b> die Umgebungsvariable LD_LIBRARY_PATH festlegen und exportieren, um den Pfad einzubeziehen.</p>	Kein Standardwert
wmqfte.file.logger.fileDirectory	Das Verzeichnis, in dem sich die Protokolldateien der Dateiprotokollfunktion befinden.	mqft/logs/coordination_dir/loggers/logger_name/logs
wmqfte.file.logger.fileSize	Gibt die Größe an, die eine Protokolldatei maximal erreichen darf. Der Wert wird in Form einer positiven ganzen Zahl größer null angegeben, gefolgt von einer der folgenden Maßeinheiten: KB, MB, GB, m (Minuten), h (Stunden), d (Tage), w (Wochen). Beispiel: wmqfte.file.logger.fileSize=5MB Gibt eine maximale Dateigröße von 5MB an. wmqfte.file.logger.fileSize=2d Legt die maximale Dateigröße auf die Daten von zwei Tagen fest.	10MB
wmqfte.file.logger.fileCount	Die Anzahl an Protokolldateien, die maximal erstellt werden. Liegt die Datenmenge über der, die maximal in dieser Anzahl an Dateien gespeichert werden kann, wird die älteste Datei gelöscht. Auf diese Weise überschreitet die Anzahl der Dateien nie den hier angegebenen Wert.	3

Tabelle 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.file.logger.mode	<p>Die Protokollfunktion arbeitet im Modus 'circular' oder 'linear'. Setzen Sie diesen Wert auf CIRCULAR oder LINEAR.</p> <p>CIRCULAR-Die Dateiprotokollfunktion schreibt Informationen in eine Datei, bis diese Datei die mit der Eigenschaft <b>wmqfte.file.logger.fileSize</b> definierte maximale Größe erreicht. Ist dies der Fall, beginnt die Dateiprotokollfunktion mit einer neuen Datei. Die maximale Anzahl Dateien, die in diesem Modus geschrieben werden, wird durch den Wert gesteuert, der über die Eigenschaft <b>wmqfte.file.logger.fileCount</b> definiert wird. Wenn die maximal zulässige Anzahl an Dateien erreicht ist, löscht die Dateiprotokollfunktion die erste Datei und erstellt sie anschließend erneut, um sie als aktive Protokolldatei einzusetzen. Wenn der in der Eigenschaft <b>wmqfte.file.logger.fileSize</b> definierte Wert eine Byteeinheit mit fester Größe (z. B. KB, MB oder GB) ist, entspricht die Obergrenze für den in diesem Modus verwendeten Plattenspeicherplatz dem Wert von <code>fileSize</code> multipliziert mit <code>fileCount</code>. Wenn der in der Eigenschaft <b>wmqfte.file.logger.fileSize</b> definierte Wert eine Zeiteinheit ist (z. B. M, H, oder W), hängt die maximale Größe vom Durchsatz der Protokollnachrichten in Ihrem System über diese Zeiträume ab. Die Namenskonvention für die Protokolldatei, die bei der Ausführung in diesem Modus verwendet wird, lautet <code>Logger_namenumber-timestamp.log</code>. Dabei gilt Folgendes:</p> <ul style="list-style-type: none"> <li>• <i>Name_der_Protokollfunktion</i> ist der Name, der der Protokollfunktion im Befehl <b>fteCreateLogger</b> zugewiesen wird.</li> <li>• <i>Nummer</i> ist die Nummer der Datei innerhalb der Gruppe.</li> <li>• <i>Zeitmarke</i> gibt den Erstellungszeitpunkt der Datei an.</li> </ul> <p>Beispiel: <code>LOGGER1-20111216123430147.log</code></p> <p>LINEAR: Die Dateiprotokollfunktion schreibt Informationen in eine Datei, bis diese Datei ihre mit der Eigenschaft <b>wmqfte.file.logger.fileSize</b> definierte maximale Größe erreicht. Ist dies der Fall, beginnt die Dateiprotokollfunktion mit einer neuen Datei. Die zuvor geschriebenen Dateien werden dabei nicht gelöscht, sodass die vorherigen Protokollnachrichten weiterhin vorliegen. Dateien werden bei Ausführung im Modus <code>Linear</code> nicht gelöscht. Daher wird die Eigenschaft <b>wmqfte.file.logger.fileCount</b> ignoriert, da es keine Obergrenze für die Anzahl der Dateien gibt, die erstellt werden können. Da es in diesem Modus keine Obergrenze gibt, muss der von den Protokolldateien belegte Speicherplatz überwacht werden, damit es zu keinen Festplattenspeicherengpässen kommt. Die Namenskonvention für die Protokolldatei, die bei der Ausführung in diesem Modus verwendet wird, lautet <code>logger_name-timestamp.log</code>. Dabei gilt Folgendes:</p> <ul style="list-style-type: none"> <li>• <i>Name_der_Protokollfunktion</i> ist der Name, der der Protokollfunktion im Befehl <b>fteCreateLogger</b> zugewiesen wird.</li> <li>• <i>Zeitmarke</i> gibt den Erstellungszeitpunkt der Datei an.</li> </ul> <p>Beispiel: <code>LOGGER-20111216123430147.log</code></p>	Kein Standardwert

Tabelle 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)


Eigenschaftsname	Beschreibung	Standardwert
wmqfte.max.retry.interval	<p>Die maximale Zeit in Sekunden zwischen Wiederholungen, wenn bei der Protokollfunktion ein permanenter Fehler auftritt.</p> <p>Einige Fehlerbedingungen (beispielsweise eine Unterbrechung der Datenbankverbindung) verhindern eine Fortsetzung der Protokollfunktion. Tritt eine solche Bedingung ein, setzt die Protokollfunktion die aktuelle Transaktion zurück, wartet eine bestimmte Zeit und wiederholt dann den Vorgang. Die Zeit, die die Protokollfunktion wartet, ist zu Beginn sehr kurz, sodass vorübergehende Fehler schnell überwunden werden können. Bei jedem neuen Versuch der Protokollfunktion allerdings verlängert sich die Wartezeit. Dies verhindert, dass zu viel unnötige Arbeit ausgeführt wird, wenn die Fehlerbedingung länger andauert, beispielsweise wenn eine Datenbank zur Wartung heruntergefahren wird.</p> <p>Legen Sie über diese Eigenschaft einen Grenzwert für die Wartezeit fest, sodass nach einer angemessenen Zeit, in der die Fehlerbedingung behoben sein sollte, eine Wiederholung stattfindet.</p>	600
 mediateShutdownTimeout	<p>Die Zeit in Sekunden, die die Protokollfunktion wartet, bis alle ausstehenden Operationen ordnungsgemäß abgeschlossen und beendet wurden. Standardmäßig wartet die Protokollfunktion 10 Sekunden darauf, dass die Operationen beendet werden. Wenn Operationen nicht vor dem Zeitlimit abgeschlossen werden, schreibt die Protokollfunktion die folgende Ereignisnachricht in output0.log und wird beendet:BFGBD0082I: The logger is ending immediately.</p> <p>Bei Angabe des Werts 'null' wartet die Protokollfunktion unbegrenzt lange auf den Abschluss aktueller Operationen.</p> <p>Der Standardwert wird verwendet, wenn der Wert von <b>mediateShutdownTimeout</b> kleiner als null ist.</p> <p>Die Eigenschaft gilt sowohl für die eigenständige Datenbankprotokollfunktion als auch für die dateibasierte Protokollfunktion.</p>	10
loggerQMgrRetryInterval	Das Intervall (in Sekunden), in dem der Prozesscontroller der Protokollfunktion prüft, ob der Warteschlangenmanager verfügbar ist.	30
maxRestartCount	Die Anzahl an Neustarts, die maximal innerhalb des Zeitintervalls möglich sind, das über die Eigenschaft 'maxRestartInterval' angegeben ist. Bei Überschreiten dieses Wertes versucht der Prozesscontroller der Protokollfunktion nicht mehr, die Protokollfunktion neu zu starten, sondern führt die über die Eigenschaft 'maxRestartDelay' vorgegebene Aktion aus.	4
maxRestartInterval	Der Zeitraum (in Sekunden), über den hinweg der Prozesscontroller der Protokollfunktion die Anzahl der bereits erfolgten Neustarts der Protokollfunktion zählt. Liegt die Anzahl der Neustart innerhalb dieses Zeitraums über dem für 'maxRestartCount' angegebenen Wert, wird die Protokollfunktion vom Prozesscontroller nicht mehr neu gestartet. Stattdessen führt der Prozesscontroller die über die Eigenschaft 'maxRestartDelay' vorgegebene Aktion aus.	120

Tabelle 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)




Eigenschaftsname	Beschreibung	Standardwert
maxRestartDelay	Gibt an, wie der Prozesscontroller der Protokollfunktion vorgeht, wenn die Anzahl der Protokollfunktionsneustarts den über die Eigenschaften 'maxRestartCount' und 'maxRestartInterval' angegebenen Wert überschreitet. Wird ein Wert kleiner oder gleich null angegeben, wird der Prozesscontroller der Protokollfunktion gestoppt. Ein Wert größer als null entspricht der Anzahl der Sekunden, nach deren Ablauf die im Prozesscontroller der Protokollfunktion gespeicherten Protokollinformationen zu den Neustarts zurückgesetzt werden und die Protokollfunktion erneut gestartet wird.	-1
wmqfte.oracle.port	Der Port, über den die Protokollfunktion eine Verbindung zur Oracle-Instanz herstellt. Dieser Port wird auch als TNS-Listener bezeichnet.	1521
wmqfte.oracle.host	Der Host, über den die Protokollfunktion eine Verbindung zur Oracle-Instanz herstellt.	localhost
armELEMTYPE	Optionale Eigenschaft. Wenn die Protokollfunktion so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMTYPE, der in der zugehörigen ARM-Richtlinie festgelegt ist. Setzen Sie ELEMTYPE für eine Protokollfunktion auf SYSBFGLG.	Nicht festgelegt
armELEMENT	Optionale Eigenschaft. Wenn die Protokollfunktion so konfiguriert ist, dass Neustarts über den Automatic Restart Manager (ARM) erfolgen, setzen Sie diese Eigenschaft auf den Wert des Parameters ARM ELEMENT, der in der zugehörigen ARM-Richtlinie festgelegt ist. Der Wert von ELEMENT kann dem Namen der Protokollfunktion entsprechen.	Nicht festgelegt
loggerQMgrAuthenticationCredentialsFile	Der Pfad der Datei, die die MQ-Berechtigungs-nachweise für die Verbindung mit dem Koordinationswarteschlangenmanager der Protokollfunktion enthält.	<p> Details zum Erstellen der Datei mit Authentifizierungsnachweisen finden Sie unter <a href="#">MQMFTCredentials.xml</a> unter z/OS konfigurieren.</p> <p> Informationen über die Position und die Berechtigungen dieser Datei finden Sie im Abschnitt <a href="#">MQMFTCredentials.xml</a> konfigurieren.</p> <p> Weitere Informationen zum Erstellen der Authentifizierungs-berechtigungsdatei finden Sie unter <a href="#">MFT- und IBM MQ-Authentifizierungsnachweis</a>.</p>
trace	<p>Optionale Eigenschaft. Tracespezifikation, wenn die Protokollfunktion beim Start mit aktiviertem Trace ausgeführt werden soll. Die Tracespezifikation ist eine durch Kommas getrennte Liste mit Klassen, dem Gleichheitszeichen und einer Tracestufe.</p> <p>Beispiel: com.ibm.wmqfte.databaselogger, com.ibm.wmqfte.databaselogger.operation=all.</p> <p>Sie können auch mehrere Tracespezifikationen in einer durch Doppelpunkte getrennten Liste angeben. Beispiel: com.ibm.wmqfte.databaselogger=mode-rate:com.ibm.wmqfte.databaselogger.operation=all</p>	--

Tabelle 41. Verbindungseigenschaften für den Bindungsmodus (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
traceFiles	Optionale Eigenschaft. Die Gesamtanzahl der zu speichernden Tracedateien. Dieser Wert gilt für den Prozesscontroller einer Protokollfunktion sowie für die Protokollfunktion selbst.	5
traceSize	Optionale Eigenschaft. Die maximale Größe (in MB) der einzelnen Tracedateien; sobald dieser Wert erreicht ist, wird der Trace in eine Folgedatei geschrieben. Dieser Wert gilt für den Prozesscontroller einer Protokollfunktion sowie für die Protokollfunktion selbst.	20
<b>V 9.1.4</b> wmqfte.file.logger.filePermissions	<p>Optionale Eigenschaft. Verwenden Sie diese Eigenschaft, um anzugeben, welche Art von Berechtigung für die Protokolldatei der Protokollfunktion erforderlich ist.</p> <p>Die Eigenschaft gilt sowohl für lineare Protokolle als auch für Umlaufprotokolle und kann die Werte <i>UserReadWriteOnly</i> oder <i>UserReadWriteAllRead</i> annehmen.</p> <p>Der Wert <i>UserReadWriteOnly</i> verfügt über die entsprechende Berechtigung 600 und der Wert <i>UserReadWriteAllRead</i> über die entsprechende Berechtigung 644.</p> <p>Jede Änderung der Berechtigung gilt für neu erstellte Dateien der Protokollfunktion.</p> <p>Wenn Sie einen ungültigen Wert für die Eigenschaft eingeben, verwendet die Protokollfunktion den Standardwert und gibt die Nachricht BFGDB0083W im Ausgabeprotokoll aus.</p>	UserReadWriteOnly

## Eigenschaften der Clientmodusverbindung

### V 9.1.0

In der folgenden Tabelle werden die zusätzlichen Eigenschaften der Protokollfunktion gezeigt, die zur Unterstützung einer Clientmodusverbindung zu einem Warteschlangenmanager für eine Protokollfunktion erforderlich sind.

Tabelle 42. Verbindungseigenschaften für den Clientmodus		
Eigenschaftsname	Beschreibung	Standardwert
wmqfte.queue.manager.host	Hostname oder IP-Adresse des Warteschlangenmanagers für die Protokollfunktion.	Kein Standardwert
wmqfte.queue.manager.port	Der Port, an dem der Warteschlangenmanager der Protokollfunktion empfangsbereit ist.	1414
wmqfte.queue.manager.channel	Name des Serververbindungskanals für den Warteschlangenmanager der Protokollfunktion.	SYSTEM.DEF.SVRCONN

Tabelle 42. Verbindungseigenschaften für den Clientmodus (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.Ssl.CipherSuite	<p>Gibt TLS-Aspekte zum Austausch von Daten zwischen der Protokollfunktion und dem Warteschlangenmanager der Protokollfunktion an.</p> <p>Der Wert von <b>wmqfte.Ssl.CipherSuite</b> ist ein CipherSuite-Name. Dieser CipherSuite-Name ist mit dem CipherSpec-Namen identisch, der im Kanal des Warteschlangenmanagers der Protokollfunktion verwendet wird.</p> <p>Weitere Informationen hierzu finden Sie im Abschnitt Namenszuordnungen von CipherSuites und CipherSpecs.</p>	Kein Standardwert
wmqfte.Ssl.PeerName	<p>Gibt den Entwurf eines definierten Namens an, der mit dem vom Warteschlangenmanager der Protokollfunktion bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Warteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.</p>	Kein Standardwert
wmqfte.Ssl.TrustStore	<p>Gibt die Position der Zertifikate an, die die Protokollfunktion akzeptiert. Der Wert von <b>wmqfte.Ssl.TrustStore</b> ist ein Dateipfad.</p> <p><b>Windows</b> Wenn es sich dabei um einen Windows-Dateipfad handelt, muss dem Backslash-Zeichen (\) ein Escapezeichen (\\) vorangestellt werden.</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.TrustStoreCredentialsFile	<p>Der Pfad zu der Datei, die den Berechtigungsnachweis <b>wmqfte.Ssl.TrustStore</b> enthält</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.TrustStoreType	<p>Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Diese Eigenschaft kann den Wert 'jks' oder 'pkcs12' haben.</p>	jks
wmqfte.Ssl.KeyStore	<p>Gibt die Position des privaten Schlüssels der Protokollfunktion an. Der Wert von <b>wmqfte.Ssl.KeyStore</b> ist ein Dateipfad.</p> <p><b>Windows</b> Wenn es sich dabei um einen Windows-Dateipfad handelt, muss dem Backslash-Zeichen (\) ein Escapezeichen (\\) vorangestellt werden.</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.KeyStore.CredentialsFile	<p>Der Pfad zu der Datei, die den Berechtigungsnachweis <b>wmqfte.Ssl.KeyStore</b> enthält</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert

Tabelle 42. Verbindungseigenschaften für den Clientmodus (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
wmqfte.Ssl.KeyStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Diese Eigenschaft kann den Wert 'jks' oder 'pkcs12' haben.	jks
wmqfte.Ssl.FipsRequired	Gibt an, dass Sie die FIPS-Unterstützung auf der Ebene der Protokollfunktion aktivieren möchten. Der Wert dieser Eigenschaft kann 'true' oder 'false' sein. Weitere Informationen finden Sie im Abschnitt <a href="#">FIPS-Unterstützung in MFT</a> .	false

### Zugehörige Verweise

„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 179

Ab IBM WebSphere MQ 7.5 können in Managed File Transfer-Eigenschaften Umgebungsvariablen für Datei- oder Verzeichnispositionen verwendet werden. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt).

„SSL-Eigenschaften für MFT“ auf Seite 233

Einige MFT-Eigenschaftendateien enthalten SSL-Eigenschaften. Mit SSL oder TLS können Sie in IBM MQ und Managed File Transfer unberechtigte Verbindungen zwischen Agenten und Warteschlangenmanagern verhindern und die Nachrichtenübertragungen zwischen Agenten und Warteschlangenmanagern verschlüsseln.

## Java-Systemeigenschaften für MFT

Einige Managed File Transfer-Befehls- und Agenteneigenschaften müssen als Java-Systemeigenschaften definiert werden, da sie die Konfiguration für ältere Funktionen bereitstellen, die die Mechanismen der Befehle und Agenteneigenschaften nicht unterstützen.

Zur Einstellung der Systemeigenschaften und anderer JVM-Optionen für eine JVM, die Managed File Transfer-Befehle ausführen muss, müssen Sie die Umgebungsvariable `BFG_JVM_PROPERTIES` definieren. Um auf einer UNIX-Plattform beispielsweise die Eigenschaft 'com.ibm.wmqfte.maxConsoleLineLength' zu setzen, ist die Variable wie folgt zu definieren:

```
export BFG_JVM_PROPERTIES="-Dcom.ibm.wmqfte.maxConsoleLineLength=132"
```

Wenn Sie einen Agenten als Windows-Dienst ausführen, können Sie dessen Java-Systemeigenschaften mit dem Parameter `-sj` im Befehl **fteModifyAgent** ändern.




Tabelle 43. Java-Systemeigenschaften		
Eigenschaftsname	Beschreibung	Wert
com.ibm.wmqfte.maxConsoleLineLength	Maximale Zeilenlänge, die in der Konsole ausgegeben werden kann. Zeilen, die diese Länge überschreiten, werden umbrochen. Der Wert wird in Byte (nicht in Zeichen) angegeben.	 Die Standardlänge für IBM i liegt bei 132 Byte.   Bei UNIX, Linux, Windows und z/OS ist die Länge unbegrenzt.



Tabelle 43. Java-Systemeigenschaften (Forts.)

Eigenschaftsname	Beschreibung	Wert
com.ibm.wmqfte.daemon.windows.windowsServiceLogFilesm	(Nur Windows). Gibt die maximale Anzahl der aufzubewahrenden Protokolldateien für Windows-Dienste an. Die Protokolldateien für Windows-Dienste werden in den Protokollverzeichnissen des Agenten und der Datenbankprotokollfunktion erstellt, wenn diese Anwendungen als Windows-Dienst ausgeführt werden. Die Dateinamen der Protokolldateien für Windows-Dienste beginnen mit dem Präfix <i>Service</i> . Die Protokolldateien enthalten Nachrichten zum Start- und Stoppvorgang des Dienstes.	5

### Zugehörige Konzepte

[MFT-Konfigurationsoptionen unter Multiplatforms](#)

[Hinweise und Tipps zur Verwendung von MFT](#)

## SSL-Eigenschaften für MFT

Einige MFT-Eigenschaftendateien enthalten SSL-Eigenschaften. Mit SSL oder TLS können Sie in IBM MQ und Managed File Transfer unberechtigte Verbindungen zwischen Agenten und Warteschlangenmanagern verhindern und die Nachrichtenübertragungen zwischen Agenten und Warteschlangenmanagern verschlüsseln.

Folgende MFT-Eigenschaftendateien enthalten SSL-Eigenschaften:

- Die Datei [agent.properties](#)
- Die Datei [coordination.properties](#)
- Die Datei [command.properties](#)
- **V 9.1.0** Die Datei [logger.properties](#)

Informationen zur Verwendung von SSL oder TLS mit Managed File Transfer finden Sie unter [SSL- oder TLS-Verschlüsselung für MFT konfigurieren](#).

Ab IBM WebSphere MQ 7.5 können Sie Umgebungsvariablen in einigen Managed File Transfer-Eigenschaften verwenden, die Datei- oder Verzeichnispositionen darstellen. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt). Weitere Informationen finden Sie unter [„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“](#) auf Seite 179.

### SSL-Eigenschaften für die Datei [agent.properties](#)

Die Datei [agent.properties](#) für einen Agenten befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name`. Zu den Eigenschaften, die darin enthalten sind, gehören die folgenden SSL-Eigenschaften:

Tabelle 44. SSL-Eigenschaften für die Datei 'agent.properties'

Eigenschaftsname	Beschreibung	Standardwert
agentSslCipherSpec	<p>Gibt das Protokoll, den Hashalgorithmus und den Verschlüsselungsalgorithmus an, die verwendet werden. Gibt außerdem an, wie viele Bit im Verschlüsselungsschlüssel verwendet werden, wenn Daten zwischen dem Agenten und dem Warteschlangenmanager des Agenten ausgetauscht werden.</p> <p>Der Wert von 'agentSslCipherSpec' ist ein CipherSpec-Name. Dieser CipherSpec-Name ist mit dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen identisch. Eine Liste gültiger CipherSpec-Namen ist in den Abschnitten <a href="#">SSL/TLS-CipherSpecs</a> und <a href="#">-CipherSuites in IBM MQ-Klassen für Java</a> und <a href="#">SSL/TLS-CipherSpecs</a> und <a href="#">-CipherSuites in IBM MQ-Klassen für JMS</a> zu finden.</p> <p>'agentSslCipherSpec' entspricht weitgehend 'agentSslCipherSuite'. Wenn sowohl 'agentSslCipherSuite' als auch 'agentSslCipherSpec' angegeben ist, wird der Wert von 'agentSslCipherSpec' verwendet.</p>	--
agentSslCipherSuite	<p>Gibt SSL-Aspekte zum Austausch von Daten zwischen dem Agenten und dem Warteschlangenmanager des Agenten an.</p> <p>Beim Wert von 'agentSslCipherSuite' handelt es sich um einen CipherSuite-Namen. Dieser CipherSuite-Name ist dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen zugeordnet. Weitere Informationen hierzu finden Sie im Abschnitt <a href="#">Namenszuordnungen von CipherSuites und CipherSpecs</a>.</p> <p>'agentSslCipherSuite' entspricht weitgehend 'agentSslCipherSpec'. Wenn sowohl 'agentSslCipherSuite' als auch 'agentSslCipherSpec' angegeben ist, wird der Wert von 'agentSslCipherSpec' verwendet.</p>	--
agentSslPeerName	<p>Gibt den Entwurf eines definierten Namens an, der mit dem vom Warteschlangenmanager des Agenten bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Warteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.</p>	--
agentSslTrustStore	<p>Gibt die Position der Zertifikate an, die der Agent akzeptiert. Beim Wert von 'agentSslTrustStore' handelt es sich um einen Dateipfad. Wenn es sich um einen Windows-Dateipfad handelt, muss dem Backslash-Zeichen (\) ein Escape-Zeichen (\\) vorangestellt werden.</p> <p>Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.</p>	--
agentSslKeyStore	<p>Gibt die Position des privaten Schlüssels des Agenten an. Beim Wert von 'agentSslKeyStore' handelt es sich um einen Dateipfad. Bei einem Windows-Dateipfad muss der umgekehrte Schrägstrich (\) mit einem Escapezeichen versehen werden (\\). Diese Eigenschaft ist nur erforderlich, wenn der Agentenwarteschlangenmanager die Clientauthentifizierung erfordert.</p> <p>Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.</p>	--
agentSslFipsRequired	<p>Gibt an, dass Sie die FIPS-Unterstützung auf Agentenebene aktivieren möchten. Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code>. Weitere Informationen finden Sie im Abschnitt <a href="#">FIPS-Unterstützung in MFT</a>.</p>	false

Tabelle 44. SSL-Eigenschaften für die Datei 'agent.properties' (Forts.)		
Eigenschaftsname	Beschreibung	Standardwert
agentSslKeyStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann jks oder pkcs12 angegeben werden.	jks
agentSslKeyStoreCredentialsFile	Der Pfad der Datei, die den agentSslKeyStore-Berechtigungs-nachweis enthält. Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert %USERPROFILE%\MQMFTCcredentials.xml unter Windows bzw. \$HOME/MQMFTCcredentials.xml auf anderen Plattformen.
agentSslTrustStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann jks oder pkcs12 angegeben werden.	jks
agentSslTrustStoreCredentialsFile	Der Pfad der Datei, die den agentSslTrustStore-Berechtigungs-nachweis enthält. Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert %USERPROFILE%\MQMFTCcredentials.xml unter Windows bzw. \$HOME/MQMFTCcredentials.xml auf anderen Plattformen.

## SSL-Eigenschaften für die Datei coordination.properties

Die Datei `coordination.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name`. Zu den Eigenschaften, die darin enthalten sind, gehören die folgenden SSL-Eigenschaften:

Tabelle 45. SSL-Eigenschaften für die Datei coordination.properties		
Eigenschaftsname	Beschreibung	Standardwert
coordinationSslCipherSpec	Gibt beim Austausch von Daten zwischen den Befehlen und dem Koordinationswarteschlangenmanager das Protokoll, den Hashalgorithmus und den Verschlüsselungsalgorithmus sowie Informationen zur Anzahl der im Verschlüsselungsschlüssel verwendeten Bits an.  Der Wert von 'coordinationSslCipherSpec' ist ein CipherSpec-Name. Dieser CipherSpec-Name ist mit dem im Kanal des Koordinationswarteschlangenmanagers verwendeten CipherSpec-Namen identisch. Eine Liste gültiger CipherSpec-Namen ist in den Abschnitten <a href="#">SSL/TLS-CipherSpecs</a> und <a href="#">-CipherSuites</a> in IBM MQ-Klassen für Java und <a href="#">SSL/TLS-CipherSpecs</a> und <a href="#">-CipherSuites</a> in IBM MQ-Klassen für JMS zu finden.  'coordinationSslCipherSpec' entspricht weitgehend 'coordinationSslCipherSuite'. Wenn sowohl 'coordinationSslCipherSuite' als auch 'coordinationSslCipherSpec' angegeben ist, wird der Wert von 'coordinationSslCipherSpec' verwendet.	--

Tabelle 45. SSL-Eigenschaften für die Datei `coordination.properties` (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<code>coordinationSslCipherSuite</code>	<p>Gibt SSL-Aspekte zum Austausch von Daten zwischen den Befehlen und dem Koordinationswarteschlangenmanager an.</p> <p>Beim Wert von '<code>coordinationSslCipherSuite</code>' handelt es sich um einen CipherSuite-Namen. Dieser CipherSuite-Namen ist dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen zugeordnet. Weitere Informationen hierzu finden Sie im Abschnitt <a href="#">Namenszuordnungen von CipherSuites und CipherSpecs</a>.</p> <p>'<code>coordinationSslCipherSuite</code>' entspricht weitgehend '<code>coordinationSslCipherSpec</code>'. Wenn sowohl '<code>coordinationSslCipherSuite</code>' als auch '<code>coordinationSslCipherSpec</code>' angegeben ist, wird der Wert von '<code>coordinationSslCipherSpec</code>' verwendet.</p>	--
<code>coordinationSslPeerName</code>	Gibt den Entwurf eines definierten Namens an, der mit dem vom Koordinationswarteschlangenmanager bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Koordinationswarteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.	--
<code>coordinationSslTrustStore</code>	<p>Gibt die Position der Zertifikate an, die die Befehle akzeptieren. Beim Wert von '<code>coordinationSslTrustStore</code>' handelt es sich um einen Dateipfad. Wenn es sich um einen Windows-Dateipfad handelt, muss dem Backslash-Zeichen (\) ein Escape-Zeichen (\\) vorangestellt werden.</p> <p>Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.</p>	--
<code>coordinationSslTrustStoreType</code>	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann <code>jks</code> oder <code>pkcs12</code> angegeben werden.	<code>jks</code>
<code>coordinationSslTrustStoreCredentialsFile</code>	<p>Der Pfad der Datei, die die <code>coordinationSslTrustStore</code>-Berechtigungsanzeige enthält.</p> <p>Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.</p>	Diese Eigenschaft hat den Standardwert <code>%USERPROFILE%\MQMFTCcredentials.xml</code> unter Windows bzw. <code>\$HOME/MQMFTCcredentials.xml</code> auf anderen Plattformen.
<code>coordinationSslKeyStore</code>	<p>Gibt die Position des privaten Schlüssels der Befehle an. Beim Wert von '<code>coordinationSslKeyStore</code>' handelt es sich um einen Dateipfad. Falls es sich um einen Windows-Dateipfad handelt, muss der umgekehrte Schrägstrich (\) mit einem Escapezeichen versehen werden (\\). Diese Eigenschaft ist nur erforderlich, wenn der Koordinationswarteschlangenmanager die Clientauthentifizierung erfordert.</p> <p>Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.</p>	--
<code>coordinationSslKeyStoreType</code>	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann <code>jks</code> oder <code>pkcs12</code> angegeben werden.	<code>jks</code>
<code>coordinationSslKeyStoreCredentialsFile</code>	<p>Der Pfad der Datei, die die <code>coordinationSslKeyStore</code>-Berechtigungsanzeige enthält.</p> <p>Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.</p>	Diese Eigenschaft hat den Standardwert <code>%USERPROFILE%\MQMFTCcredentials.xml</code> unter Windows bzw. <code>\$HOME/MQMFTCcredentials.xml</code> auf anderen Plattformen.

Tabelle 45. SSL-Eigenschaften für die Datei `coordination.properties` (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<code>coordinationSslFipsRequired</code>	Gibt an, dass Sie die FIPS-Unterstützung auf Ebene des Koordinationswarteschlangenmanagers aktivieren möchten. Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code> . Weitere Informationen finden Sie im Abschnitt <a href="#">FIPS-Unterstützung in MFT</a> .	<code>false</code>

## SSL-Eigenschaften für die Datei `command.properties`

Die Datei `command.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name`. Zu den Eigenschaften, die darin enthalten sind, gehören die folgenden SSL-Eigenschaften:

Tabelle 46. SSL-Eigenschaften für die Datei `command.properties`

Eigenschaftsname	Beschreibung	Standardwert
<code>connectionSslCipherSpec</code>	Gibt beim Austausch von Daten zwischen den Befehlen und dem Befehlswarteschlangenmanager das Protokoll, den Hashalgorithmus und den Verschlüsselungsalgorithmus sowie Informationen zur Anzahl der im Verschlüsselungsschlüssel verwendeten Bits an.  Beim Wert von 'connectionSslCipherSpec' handelt es sich um einen CipherSpec-Namen. Dieser CipherSpec-Name ist mit dem im Kanal des Befehlswarteschlangenmanagers verwendeten CipherSpec-Namen identisch. Eine Liste gültiger CipherSpec-Namen ist in <a href="#">SSL/TLS-CipherSpecs</a> und <a href="#">-CipherSuites</a> in IBM MQ-Klassen für Java und <a href="#">SSL/TLS-CipherSpecs</a> und <a href="#">-CipherSuites</a> in IBM MQ-Klassen für JMS zu finden.  'connectionSslCipherSpec' entspricht weitgehend 'connectionSslCipherSuite'. Wenn sowohl 'connectionSslCipherSuite' als auch 'connectionSslCipherSpec' angegeben ist, wird der Wert von 'connectionSslCipherSpec' verwendet.	--
<code>connectionSslCipherSuite</code>	Gibt SSL-Aspekte zum Austausch von Daten zwischen den Befehlen und dem Warteschlangenmanager für den Befehl an.  Beim Wert von 'connectionSslCipherSuite' handelt es sich um einen CipherSuite-Namen. Dieser CipherSuite-Name ist dem im Kanal des Agentenwarteschlangenmanagers verwendeten CipherSpec-Namen zugeordnet. Weitere Informationen hierzu finden Sie im Abschnitt <a href="#">Namenszuordnungen von CipherSuites</a> und <a href="#">CipherSpecs</a> .  'connectionSslCipherSuite' entspricht weitgehend 'connectionSslCipherSpec'. Wenn sowohl 'connectionSslCipherSuite' als auch 'connectionSslCipherSpec' angegeben ist, wird der Wert von 'connectionSslCipherSpec' verwendet.	--
<code>connectionSslPeerName</code>	Gibt den Entwurf eines definierten Namens an, der mit dem vom Befehlswarteschlangenmanager bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Befehlswarteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.	--
<code>connectionSslTrustStore</code>	Gibt die Position der Zertifikate an, die die Befehle akzeptieren. Beim Wert von 'connectionSslTrustStore' handelt es sich um einen Dateipfad. Wenn es sich um einen Windows-Dateipfad handelt, muss dem Backslash-Zeichen (\) ein Escape-Zeichen (\\) vorangestellt werden.  Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	--
<code>connectionSslTrustStoreType</code>	Die Art des zu verwendenden SSL-Truststores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann <code>jks</code> oder <code>pkcs12</code> angegeben werden.	<code>jks</code>

Tabelle 46. SSL-Eigenschaften für die Datei `command.properties` (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
<code>connectionSslTrustStoreCredentialsFile</code>	Der Pfad der Datei, die die <code>connectionSslTrustStore</code> -Berechtigungs-nachweise enthält. Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert <code>%USERPROFILE%\MQMFTCcredentials.xml</code> unter Windows bzw. <code>\$HOME/MQMFTCcredentials.xml</code> auf anderen Plattformen.
<code>connectionSslKeyStore</code>	Gibt die Position des privaten Schlüssels der Befehle an. Beim Wert von <code>'connectionSslKeyStore'</code> handelt es sich um einen Dateipfad. Falls es sich um einen Windows-Dateipfad handelt, muss der umgekehrte Schrägstrich ( <code>\</code> ) mit einem Escapezeichen versehen werden ( <code>\\</code> ). Diese Eigenschaft ist nur erforderlich, wenn der Befehlswarteschlangenmanager die Clientauthentifizierung erfordert. Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	--
<code>connectionSslKeyStoreType</code>	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Für diese Eigenschaft kann <code>jks</code> oder <code>pkcs12</code> angegeben werden. Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	<code>jks</code>
<code>connectionSslKeyStoreCredentialsFile</code>	Der Pfad der Datei, die die <code>connectionSslKeyStore</code> -Berechtigungs-nachweise enthält. Ab IBM WebSphere MQ 7.5 kann der Wert dieser Eigenschaft Umgebungsvariablen enthalten.	Diese Eigenschaft hat den Standardwert <code>%USERPROFILE%\MQMFTCcredentials.xml</code> unter Windows bzw. <code>\$HOME/MQMFTCcredentials.xml</code> auf anderen Plattformen.
<code>connectionSslFipsRequired</code>	Gibt an, dass Sie die FIPS-Unterstützung auf Ebene des Befehlswarteschlangenmanagers aktivieren möchten. Mögliche Werte für diese Eigenschaft sind <code>true</code> oder <code>false</code> . Weitere Informationen finden Sie im Abschnitt <a href="#">FIPS-Unterstützung in MFT</a> .	<code>false</code>

## SSL-Eigenschaften für die Datei `logger.properties`

V 9.1.0

Die Datei `logger.properties` befindet sich im Verzeichnis `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name`. Zu den Eigenschaften, die für die Unterstützung der Clientmodusverbindung mit einem Protokollfunktions-Warteschlangenmanager erforderlich sind, gehören folgende SSL-Eigenschaften:

Tabelle 47. SSL-Eigenschaften für die Datei `'logger.properties'`

Eigenschaftsname	Beschreibung	Standardwert
<code>wmqfte.queue.manager.host</code>	Hostname oder IP-Adresse des Warteschlangenmanagers für die Protokollfunktion.	Kein Standardwert
<code>wmqfte.queue.manager.port</code>	Der Port, an dem der Warteschlangenmanager der Protokollfunktion empfangsbereit ist.	1414
<code>wmqfte.queue.manager.channel</code>	Name des Serververbindungskanals für den Warteschlangenmanager der Protokollfunktion.	<code>SYSTEM.DEF.SVRCONN</code>

Tabelle 47. SSL-Eigenschaften für die Datei 'logger.properties' (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.Ssl.CipherSuite	<p>Gibt TLS-Aspekte zum Austausch von Daten zwischen der Protokollfunktion und dem Warteschlangenmanager der Protokollfunktion an.</p> <p>Der Wert von <b>wmqfte.Ssl.CipherSuite</b> ist ein CipherSuite-Name. Dieser CipherSuite-Name ist mit dem CipherSpec-Namen identisch, der im Kanal des Warteschlangenmanagers der Protokollfunktion verwendet wird.</p> <p>Weitere Informationen hierzu finden Sie im Abschnitt Namenszuordnungen von CipherSuites und CipherSpecs.</p>	Kein Standardwert
wmqfte.Ssl.PeerName	<p>Gibt den Entwurf eines definierten Namens an, der mit dem vom Warteschlangenmanager der Protokollfunktion bereitgestellten Namen übereinstimmen muss. Mit dem definierten Namen wird das vom Warteschlangenmanager bei der Verbindung bereitgestellte Zertifikat für die Identifizierung geprüft.</p>	Kein Standardwert
wmqfte.Ssl.TrustStore	<p>Gibt die Position der Zertifikate an, die die Protokollfunktion akzeptiert. Der Wert von <b>wmqfte.Ssl.TrustStore</b> ist ein Dateipfad.</p> <p><b>Windows</b> Wenn es sich dabei um einen Windows-Dateipfad handelt, muss dem Backslash-Zeichen (\) ein Escapezeichen (\\) vorangestellt werden.</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.TrustStoreCredentialsFile	<p>Der Pfad zu der Datei, die den Berechtigungsnachweis <b>wmqfte.Ssl.TrustStore</b> enthält</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.TrustStoreType	<p>Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Diese Eigenschaft kann den Wert 'jks' oder 'pkcs12' haben.</p>	jks
wmqfte.Ssl.KeyStore	<p>Gibt die Position des privaten Schlüssels der Protokollfunktion an. Der Wert von <b>wmqfte.Ssl.KeyStore</b> ist ein Dateipfad.</p> <p><b>Windows</b> Wenn es sich dabei um einen Windows-Dateipfad handelt, muss dem Backslash-Zeichen (\) ein Escapezeichen (\\) vorangestellt werden.</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert
wmqfte.Ssl.KeyStore.CredentialsFile	<p>Der Pfad zu der Datei, die den Berechtigungsnachweis <b>wmqfte.Ssl.KeyStore</b> enthält</p> <p>Beachten Sie, dass der Wert dieser Eigenschaft Umgebungsvariablen enthalten kann.</p>	Kein Standardwert

Tabelle 47. SSL-Eigenschaften für die Datei 'logger.properties' (Forts.)

Eigenschaftsname	Beschreibung	Standardwert
wmqfte.Ssl.KeyStoreType	Die Art des zu verwendenden SSL-Keystores. Sowohl JKS- als auch PKCS#12-Truststores werden unterstützt. Diese Eigenschaft kann den Wert 'jks' oder 'pkcs12' haben.	jks
wmqfte.Ssl.FipsRequired	Gibt an, dass Sie die FIPS-Unterstützung auf der Ebene der Protokollfunktion aktivieren möchten. Der Wert dieser Eigenschaft kann 'true' oder 'false' sein. Weitere Informationen finden Sie im Abschnitt <a href="#">FIPS-Unterstützung in MFT</a> .	false

## SHA-2-CipherSpecs und -CipherSuites für MFT

Managed File Transfer unterstützt SHA-2-CipherSpecs und -CipherSuites.

Weitere Informationen zu den für Verbindungen zwischen Agenten und IBM MQ-Warteschlangenmanagern verfügbaren CipherSpecs und CipherSuites finden Sie im Abschnitt [SSL CipherSpecs and CipherSuites](#).

Weitere Informationen zur Konfiguration von CipherSpecs und CipherSuites für die Verwendung mit dem Protokollbridgeagenten und FTPS-Servern finden Sie in den Abschnitten [Unterstützung für den FTPS-Server durch die Protokollbridge](#) und [Format der Eigenschaftendatei für die Protokollbridge](#).

Für die Konformität mit SP 800-131A wird Folgendes vorausgesetzt:

- Sie benötigen ein entsprechend konfiguriertes FTPS; SFTP wird nicht unterstützt.
- Der Remote Server darf nur SP 800-131A-konforme Cipher-Suites senden.

### Zugehörige Verweise

„SSL-Eigenschaften für MFT“ auf Seite 233

Einige MFT-Eigenschaftendateien enthalten SSL-Eigenschaften. Mit SSL oder TLS können Sie in IBM MQ und Managed File Transfer unberechtigte Verbindungen zwischen Agenten und Warteschlangenmanagern verhindern und die Nachrichtenübertragungen zwischen Agenten und Warteschlangenmanagern verschlüsseln.

## Konfigurationsdateien der MFT-Dateiprotokollfunktion

Neben der Datei `logger.properties` enthält das Konfigurationsverzeichnis einer eigenständigen Managed File Transfer-Dateiprotokollfunktion noch eine XML-Konfigurationsdatei (`FileLoggerFormat.xml`), in der das Format definiert ist, in dem die Dateiprotokollfunktion Nachrichten in die Protokolldatei schreibt. Der Inhalt dieser Datei muss dem in der Datei `FileLoggerFormat.xsd` definierten XML-Schema entsprechen.

### Standardprotokollformat der eigenständigen MFT-Dateiprotokollfunktion

Definition des Standardprotokolldateiformats für die eigenständige Managed File Transfer-Dateiprotokollfunktion.

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  version="1.00" xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <callCompleted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
          <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="3" ignoreNull="false">/transaction/status/@resultCode</insert>
          <insert type="user" width="0" ignoreNull="false">/transaction/agent/@agent</insert>
          <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMgr</insert>
          <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/command/@ty|
```



```

pe</insert>
  <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/com
mand/@name</insert>
  <insert type="system" width="0" ignoreNull="true">callArguments</insert>
  <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/callRe
sult/@outcome</insert>
  <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/callResult/re
sult/error</insert>
</inserts>
<separator>;</separator>
</format>
</callCompleted>
<callStarted>
<format>
<inserts>
  <insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
  <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
  <insert type="system" width="6" ignoreNull="false">type</insert>
  <insert type="user" width="0" ignoreNull="false">/transaction/agent/@agent</insert>
  <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMgr</insert>
  <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
  <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/call/command/@ty
pe</insert>
mand/@name</insert>
  <insert type="system" width="0" ignoreNull="true">callArguments</insert>
</inserts>
<separator>;</separator>
</format>
</callStarted>
<monitorAction>
<format>
<inserts>
  <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
  <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
  <insert type="system" width="6" ignoreNull="false">type</insert>
  <insert type="user" width="3" ignoreNull="false">/monitorLog/status/@resultCode</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
</inserts>
<separator>;</separator>
</format>
</monitorAction>
<monitorCreate>
<format>
<inserts>
  <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
  <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
  <insert type="system" width="6" ignoreNull="false">type</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
</inserts>
<separator>;</separator>
</format>
</monitorCreate>
<monitorFired>
<format>
<inserts>
  <insert type="user" width="19" ignoreNull="false">/monitorLog/action/@time</insert>
  <insert type="user" width="48" ignoreNull="false">/monitorLog/@referenceId</insert>
  <insert type="system" width="6" ignoreNull="false">type</insert>
  <insert type="user" width="3" ignoreNull="false">/monitorLog/status/@resultCode</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/@monitorName</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@agent</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/monitorAgent/@QMgr</insert>
  <insert type="user" width="0" ignoreNull="false">/monitorLog/action</insert>
  <insert type="user" width="48" ignoreNull="false">/monitorLog/references/taskRequest</insert>
</inserts>
<separator>;</separator>
</format>
</monitorFired>
<notAuthorized>
<format>
<inserts>
  <insert type="user" width="19" ignoreNull="false">/notAuthorized/action/@time</insert>
  <insert type="user" width="48" ignoreNull="false">/notAuthorized/@ID</insert>
  <insert type="system" width="6" ignoreNull="false">type</insert>
  <insert type="user" width="3" ignoreNull="false">/notAuthorized/status/@resultCode</insert>

```



```

<insert type="system" width="10" ignoreNull="false">type</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@QMgr |
/transaction/sourceWebUser/@webGatewayAgentQMgr |
/transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@QMgr |
/transaction/destinationWebUser/@webGatewayAgentQMgr |
/transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
</inserts>
<separator>;</separator>
</format>
</scheduleSubmitTransfer>
<scheduleSubmitTransferSet>
<format>
<inserts>
<insert type="user" width="19" ignoreNull="false">/schedulelog/action/@time</insert>
<insert type="user" width="48" ignoreNull="false">/schedulelog/@ID</insert>
<insert type="system" width="10" ignoreNull="false">type</insert>
<insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
<insert type="user" width="5" ignoreNull="true">source/@type</insert>
<insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
<insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</in
sert>
<insert type="user" width="5" ignoreNull="true">destination/@type</insert>
<insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
</inserts>
<separator>;</separator>
</format>
</scheduleSubmitTransferSet>
<transferStarted>
<format>
<inserts>
<insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
<insert type="system" width="6" ignoreNull="false">type</insert>
<insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMgr |
/transaction/sourceWebUser/@webGatewayAgentQMgr |
/transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
/transaction/sourceWebUser/@webGatewayAgentType |
/transaction/sourceWebGateway/@webGatewayAgentType</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/destinationAgent/@QMgr |
/transaction/destinationWebUser/@webGatewayAgentQMgr |
/transaction/destinationWebGateway/@webGatewayAgentQMgr</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/originator/userID</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/job/name</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/scheduleLog/@ID</insert>
</inserts>
<separator>;</separator>
</format>
</transferStarted>
<transferCancelled>
<format>
<inserts>
<insert type="user" width="19" ignoreNull="false">/transaction/action/@time</insert>
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
<insert type="system" width="6" ignoreNull="false">type</insert>
<insert type="user" width="3" ignoreNull="true">/transaction/status/@resultCode</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/sourceAgent/@agent |
/transaction/sourceWebUser/@webGatewayAgentName |
/transaction/sourceWebGateway/@webGatewayAgentName</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@QMgr |
/transaction/sourceWebUser/@webGatewayAgentQMgr |
/transaction/sourceWebGateway/@webGatewayAgentQMgr</insert>
<insert type="user" width="0" ignoreNull="true">/transaction/sourceAgent/@agentType |
/transaction/sourceWebUser/@webGatewayAgentType |
/transaction/sourceWebGateway/@webGatewayAgentType</insert>
<insert type="user" width="0" ignoreNull="false">/transaction/destinationAgent/@agent |
/transaction/destinationWebUser/@webGatewayAgentName |
/transaction/destinationWebGateway/@webGatewayAgentName</insert>

```



```

<insert type="system" width="6" ignoreNull="false">type</insert>
<insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
<insert type="user" width="0" ignoreNull="false">source/file | source/queue</insert>
<insert type="user" width="0" ignoreNull="false">source/file/@size | source/queue/@size</in
sert>
<insert type="user" width="5" ignoreNull="true">source/@type</insert>
<insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
<insert type="user" width="0" ignoreNull="true">source/file/@alias | source/queue/@ali
as</insert>
<insert type="user" width="0" ignoreNull="true">source/file/@fileSpace | source/queue/@file
space</insert>
<insert type="user" width="0" ignoreNull="true">source/@correlationBoolean1</insert>
<insert type="user" width="0" ignoreNull="true">source/@correlationNum1</insert>
<insert type="user" width="0" ignoreNull="true">source/@correlationString1</insert>
<insert type="user" width="0" ignoreNull="false">destination/file | destination/queue</in
sert>
<insert type="user" width="0" ignoreNull="false">destination/file/@size | destination/queue/
@size</insert>
<insert type="user" width="5" ignoreNull="true">destination/@type</insert>
<insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
<insert type="user" width="0" ignoreNull="true">destination/file/@alias | destination/queue/
@alias</insert>
<insert type="user" width="0" ignoreNull="true">destination/file/@fileSpace | destinati
on/queue/@fileSpace</insert>
<insert type="user" width="0" ignoreNull="true">destination/file/@truncateRecords</insert>
<insert type="user" width="0" ignoreNull="true">destination/@correlationBoolean1</insert>
<insert type="user" width="0" ignoreNull="true">destination/@correlationNum1</insert>
<insert type="user" width="0" ignoreNull="true">destination/@correlationString1</insert>
<insert type="user" width="0" ignoreNull="true">status/supplement</insert>
</inserts>
<separator>;</separator>
</format>
</transferProgress>
</messageTypes>
</logFormatDefinition>

```

## Zugehörige Verweise

[Format der eigenständigen MFT-Dateiprotokollfunktion](#)

„XSD-Format (eigenständiges Dateiprotokollfunktionsformat)” auf Seite 245

Das Schemaformat einer eigenständigen Dateiprotokollfunktion.

## XSD-Format (eigenständiges Dateiprotokollfunktionsformat)

Das Schemaformat einer eigenständigen Dateiprotokollfunktion.

## Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
@start_non_restricted_prolog@
Version: %Z% %I% %W% %E% %U% [%H% %T%]

Licensed Materials - Property of IBM

5724-H72

Copyright IBM Corp. 2011, 2024. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with
IBM Corp.
@end_non_restricted_prolog@
-->
<!--
This schema defines the format of the FileLoggerFormat XML file that contains the definition
of the format to use when logging FTE log messages to a file. When an XML file that conforms
to this schema is processed by a file logger it can contain definitions for one or more
message type(s) that define how log messages of those types are output to the file log.
-->

<xsd:schema xmlns:xsd="https://www.w3.org/2001/XMLSchema">
<xsd:include schemaLocation="fteutils.xsd"/>

<!--

```

```

    Defines the logFileDefinition and version number
    <logFileDefinition version="1.00" ...
      <messageTypes>
        ...
      </messageTypes>
    </logFileDefinition>
  -->
<xsd:element name="logFileDefinition">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="messageTypes" type="messageTypesType" maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="version" type="versionType" use="required"/>
  </xsd:complexType>
</xsd:element>

<!--
  Defines the set of accepted message types. The definition of individual message types
  is optional. If a particular types element is present but empty then no line will be
  output for messages of that type. If a particular types element is not present then
  the default format will be used to format messages of that type.
-->
<xsd:complexType name="messageTypesType">
  <xsd:sequence>
    <xsd:element name="callCompleted" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="callStarted" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="monitorAction" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="monitorCreate" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="monitorFired" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="notAuthorized" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="scheduleDelete" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="scheduleExpire" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="scheduleSkipped" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="scheduleSubmitInfo" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="scheduleSubmitTransfer" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="scheduleSubmitTransferSet" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="transferStarted" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="transferCancelled" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="transferComplete" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="transferDelete" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
    <xsd:element name="transferProgress" type="messageType" maxOccurs="1" minOccurs="0"
curs="0" />
  </xsd:sequence>
</xsd:complexType>

<!--
  Defines the content of a message type definition e.g.

  <callStarted>
    <format>
      ...
    </format>
  </callStarted>
-->
<xsd:complexType name="messageType">
  <xsd:sequence>
    <xsd:element name="format" type="messageFormatType" maxOccurs="1" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<!--
  Defines the content of a message format definition e.g.

  <format>
    <inserts>
      ...

```

```

        </inserts
        <separator>;</separator>
    </format>
-->
<xsd:complexType name="messageFormatType">
    <xsd:sequence>
        <xsd:element name="inserts" type="insertsType" maxOccurs="1" minOccurs="1"/>
        <xsd:element name="separator" type="scheduleType" maxOccurs="1" minOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<!--
    Defines the content of the inserts element e.g.

    <inserts>
    <insert ...>
    <insert ...>
    ..
    </inserts>
-->
<xsd:complexType name="insertsType">
    <xsd:sequence>
        <xsd:element name="insert" type="insertType" maxOccurs="unbounded" minOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<!--
    Defines the content of an insert definition e.g.

    <insert type="user" width="0" ignoreNull="true">/transaction/@ID</insert>
-->
<xsd:complexType name="insertType">
    <xsd:attribute name="type" type="insertTypeType" use="required"/>
    <xsd:attribute name="width" type="xsd:nonNegativeInteger" use="required"/>
    <xsd:attribute name="ignoreNull" type="xsd:boolean" use="required"/>
</xsd:complexType>

<!--
    Defines the accepted choices for the insert type attribute.
-->
<xsd:simpleType name="insertTypeType">
    <xsd:restriction base="xsd:token">
        <xsd:enumeration value="user"/>
        <xsd:enumeration value="system"/>
    </xsd:restriction>
</xsd:simpleType>

</xsd:schema>

```

### Zugehörige Verweise

[Format der eigenständigen MFT-Dateiprotokollfunktion](#)

„Standardprotokollformat der eigenständigen MFT-Dateiprotokollfunktion“ auf Seite 240

Definition des Standardprotokolldateiformats für die eigenständige Managed File Transfer-Dateiprotokollfunktion.

## Bibliothek SBFGCMD5

Die von IBM MQ Managed File Transfer for z/OS bereitgestellte Bibliothek SBFGCMD5 enthält Member, die als Vorlagen für Jobs dienen, welche zum Erstellen einer Managed File Transfer-Konfiguration und zum Erstellen und Verwalten eines Agenten oder einer Protokollfunktion verwendet werden können.

Die Inhalte der Bibliothek werden in der folgenden Tabelle angezeigt.

Teildatei	Beschreibung
BFGCOPY	Job, mit dem eine Kopie der Bibliothek SBFGCMD5 erstellt wird
BFGCUSTM	Job, mit dem eine Kopie der Bibliothek für einen Agenten oder eine Protokollfunktion angepasst wird
BFGXCROB	Beispielvorlage <b>fteObfuscate</b>
BFGXLGCR	Vorlage <b>fteCreateLogger</b>

<b>Teildatei</b>	<b>Beschreibung</b>
BFGXMNCR	Beispielvorlage <b>fteCreateMonitor</b>
BFGXMNDE	Beispielvorlage <b>fteDeleteMonitor</b>
BFGXPRAN	Beispielvorlage <b>fteAnt</b>
BFGXSTDE	Beispielvorlage <b>fteDeleteScheduledTransfer</b>
BFGXTMCR	Beispielvorlage <b>fteCreateTemplate</b>
BFGXTMDE	Beispielvorlage <b>fteDeleteTemplate</b>
BFGXTRCA	Beispielvorlage <b>fteCancelTransfer</b>
BFGXTRCR	Beispielvorlage <b>fteCreateTransfer</b>
BFGYAGST	Vorlage für eine gestartete Taskprozedur zum Starten eines Agenten
BFGYLGST	Vorlage für eine gestartete Taskprozedur zum Starten einer Protokollfunktion
BFGZAGCL	Beispielvorlage <b>fteCleanAgent</b>
BFGZAGCR	Beispielvorlage <b>fteCreateAgent</b>
BFGZAGDE	Beispielvorlage <b>fteDeleteAgent</b>
BFGZAGLG	Beispielvorlage <b>fteSetAgentLogLevel</b>
BFGZAGLI	Beispielvorlage <b>fteListAgents</b>
BFGZAGMG	Beispielvorlage <b>fteMigrateAgent</b>
BFGZAGPI	Beispielvorlage <b>ftePingAgent</b>
BFGZAGSH	Beispielvorlage <b>fteShowAgentDetails</b>
BFGZAGSP	Beispielvorlage <b>fteStopAgent</b>
BFGZAGST	Beispielvorlage <b>fteStartAgent</b>
BFGZAGTC	Beispielvorlage <b>fteSetAgentTraceLevel</b>
BFGZCFCR	Beispielvorlage <b>fteSetupCoordination</b>
BFGZCFDF	Beispielvorlage <b>fteChangeDefaultConfigurationOptions</b>
BFGZCMCR	Beispielvorlage <b>fteSetupCommands</b>
BFGZCMD	Schablone für REXX-Script, das von anderen Mitgliedern in der Datei verwendet wird.
BFGZLGDE	Beispielvorlage <b>fteDeleteLogger</b>
BFGZLGMG	Beispielvorlage <b>fteMigrateLogger</b>
BFGZLGSH	Beispielvorlage <b>fteShowLoggerDetails</b>
BFGZLGSP	Beispielvorlage <b>fteStopLogger</b>
BFGZLGST	Beispielvorlage <b>fteStartLogger</b>
BFGZLGTC	Beispielvorlage <b>fteSetLoggerTraceLevel</b>
BFGZMCLI	Beispielvorlage <b>fteListMonitors</b>
BFGZPID	Beispielvorlage <b>fteSetProductId</b>
BFGZPROF	Schablone für Shell-Script, das von anderen Mitgliedern in der Datei verwendet wird.



Teildatei	Beschreibung
BFGZPRSH	Beispielvorlage <b>fteDisplayVersion</b>
BFGZRAS	Beispielvorlage <b>fteRas</b>
BFGZSTLI	Beispielvorlage <b>fteListScheduledTransfers</b>
BFGZTMLI	Beispielvorlage <b>fteListTemplates</b>

Weitere Einzelheiten zur Verwendung der Bibliothek SBFGCMD5, um eine neue Bibliothek zum Erstellen einer Managed File Transfer-Konfiguration und zum Erstellen und Verwalten eines Agenten oder einer Protokollfunktion zu generieren, finden Sie im Abschnitt [Managed File Transfer for z/OS konfigurieren](#).

### Zugehörige Verweise

„Verwendung von Umgebungsvariablen in MFT-Eigenschaften“ auf Seite 179

Ab IBM WebSphere MQ 7.5 können in Managed File Transfer-Eigenschaften Umgebungsvariablen für Datei- oder Verzeichnispositionen verwendet werden. Dadurch passen sich die Verzeichnis- oder Dateipfade bei der Ausführung von Teilen des Produkts an Umgebungsänderungen an (z. B. an den Benutzer, der den Prozess ausführt).

## Thema 'SYSTEM.FTE'

Das SYSTEM.FTE ist ein Thema im Koordinationswarteschlangenmanager, das von Managed File Transfer verwendet wird, um Übertragungen zu protokollieren und Informationen zu Agenten, Überwachungen, Zeitplänen und Vorlagen zu speichern.

### Themenstruktur

```
SYSTEM.FTE
  /Agents
    /agent_name
  /monitors
    /agent_name
  /Scheduler
    /agent_name
  /Templates
    /template_ID
  /Transfers
    /agent_name
    /transfer_ID
  /Log
    /agent_name
    /Monitors
    /schedule_ID
    /transfer_ID
```

#### SYSTEM.FTE/Agents/Agentenname

Dieser Abschnitt enthält eine ständige Veröffentlichung mit der Beschreibung eines Agenten im Managed File Transfer-Netz und dessen Eigenschaften. Die Nachricht zu diesem Thema wird regelmäßig entsprechend dem Agentenstatus aktualisiert. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Agentenstatusnachricht](#).

#### SYSTEM.FTE/monitors/Agentenname

Dieses Thema enthält eine ständige Veröffentlichung mit der Beschreibung der Ressourcenüberwachungen, die dem Agenten *Agentenname* zugeordnet sind. Die XML-Struktur der ständigen Veröffentlichung entspricht dem Schema `MonitorList.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Überwachungslistennachricht](#).

#### SYSTEM.FTE/Scheduler/Agentenname

Dieses Thema enthält eine ständige Veröffentlichung mit der Beschreibung aller aktiven Zeitpläne, die dem Agenten *Agentenname* zugeordnet sind. Die XML-Struktur der ständigen Veröffentlichung entspricht dem Schema `ScheduleList.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Planungslistennachricht](#).

### SYSTEM.FTE/Templates

Dieses Thema enthält eine ständige Veröffentlichung mit einer Beschreibung aller Vorlagen, die in Ihrer Managed File Transfer-Topologie definiert sind.

- Die Veröffentlichung, die jeder Vorlage zugeordnet ist, wird in einem Unterthema mit dem Namen `SYSTEM.FTE/Templates/template_ID` veröffentlicht.

Ein Beispiel für den Inhalt dieser ständigen Veröffentlichung finden Sie im Abschnitt [MFT-Beispielvorlage XML-Nachricht](#).

### SYSTEM.FTE/Transfers/Agentenname

Dieses Thema enthält Veröffentlichungen mit der Beschreibung des Status von Übertragungen, deren Ausgangspunkt der Agent *Agentenname* ist. Die Veröffentlichungen, die jeder Übertragung zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Transfers/agent_name/transfer_ID` veröffentlicht. Mit diesen Veröffentlichungen stellt das IBM MQ Explorer-Plug-in Informationen zum Fortschritt der einzelnen Übertragungen bereit. Die XML-Struktur der Veröffentlichung entspricht dem Schema `TransferStatus.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für Dateiübertragungsstatusnachricht](#).

### SYSTEM.FTE/Log/Agentenname

Dieses Thema enthält Veröffentlichungen, in denen Informationen zu Übertragungen, Überwachungen und Zeitplänen protokolliert sind, deren Ausgangspunkt der Agent *Agentenname* ist. Diese Veröffentlichungen können von der Datenbankprotokollfunktion aufgezeichnet werden; damit stehen Überwachungsdatensätze zu den im Managed File Transfer-Netz auftretenden Ereignissen zur Verfügung.

- Die Veröffentlichungen, die jeder Übertragung zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Log/agent_name/transfer_ID` veröffentlicht und die XML der Veröffentlichung entspricht dem Schema `TransferLog.xsd`. Weitere Informationen finden Sie im Abschnitt [Formate für Dateiübertragungsprotokollnachricht](#).
- Die Veröffentlichungen, die jeder geplanten Übertragung zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Log/agent_name/schedule_ID` veröffentlicht und die XML der Veröffentlichung entspricht dem Schema `ScheduleLog.xsd`. Weitere Informationen finden Sie im Abschnitt [Formate für geplante Dateiübertragungsprotokollnachricht](#).
- Die Veröffentlichungen, die den einzelnen Überwachungen zugeordnet sind, werden in einem Unterthema mit dem Namen `SYSTEM.FTE/Log/agent_name/Monitors/monitor_name/monitor_ID` veröffentlicht und die XML der Veröffentlichung entspricht dem Schema `MonitorLog.xsd`. Weitere Informationen finden Sie im Abschnitt [Format für MFT-Überwachungsprotokollnachricht](#).

## Einstellungen von MFT-Agentenwarteschlangen

Die über den Befehl `fteCreateAgent` generierten MQSC-Befehlsscripts erstellen die Agentenwarteschlangen, wobei die Parameter auf die folgenden Werte gesetzt sind. Wenn Sie die zur Erstellung der Warteschlangen bereitgestellten MQSC-Scripts nicht verwenden, sondern die Warteschlangen manuell erstellen, müssen die folgenden Parameter unbedingt auf die angegebenen Werte gesetzt werden.

### Betriebswarteschlangen des Agenten

Die Betriebswarteschlangen des Agenten haben folgende Namen:

- `SYSTEM.FTE.COMMAND.Agentenname`
- `SYSTEM.FTE.DATA.Agentenname`
- `SYSTEM.FTE.EVENT.Agentenname`
- `SYSTEM.FTE.REPLY.Agentenname`
- `SYSTEM.FTE.STATE.Agentenname`

Parameter	Wert (falls anwendbar)
DEFPRTY	0

<i>Tabelle 48. Parameter für die Betriebswarteschlangen des Agenten (Forts.)</i>	
<b>Parameter</b>	<b>Wert (falls anwendbar)</b>
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	5000
MAXMSGL	4194304
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	NORMAL
REPLACE	

### **Berechtigungswarteschlangen des Agenten**

Die Berechtigungswarteschlangen des Agenten haben folgende Namen:

- SYSTEM.FTE.AUTHADM1.*Agentenname*
- SYSTEM.FTE.AUTHAGT1. *Agentenname*
- SYSTEM.FTE.AUTHMON1.*Agentenname*
- SYSTEM.FTE.AUTHOPS1.*Agentenname*
- SYSTEM.FTE.AUTHSCH1.*Agentenname*
- SYSTEM.FTE.AUTHTRN1.*Agentenname*

<i>Tabelle 49. Parameter für die Berechtigungswarteschlangen des Agenten</i>	
<b>Parameter</b>	<b>Wert (falls anwendbar)</b>
DEFPRTY	0
DEFSOPT	SHARED
GET	ENABLED
MAXDEPTH	0
MAXMSGL	0
MSGDLVSQ	PRIORITY
PUT	ENABLED
RETINTVL	999999999
SHARE	
NOTRIGGER	
USAGE	NORMAL
REPLACE	

## Zugehörige Verweise

[fteCreateAgent](#)

## MFT-Systemwarteschlangen und der Systemabschnitt

Managed File Transfer weist zahlreiche Systemwarteschlangen und einen Systemabschnitt auf, die nur für die interne Verwendung vorgesehen sind.

Alle Warteschlangen, deren Name mit SYSTEM.FTE beginnt, sind interne Systemwarteschlangen für Managed File Transfer (MFT). Löschen Sie diese Warteschlangen nicht, da dies die ordnungsgemäße Funktion von IBM MQ MFT verhindert. [Tabelle 50 auf Seite 252](#) zeigt, welcher Nachrichtentyp sich in jeder Warteschlange befindet:

<b>Warteschlangenname</b>	<b>Warteschlangentyp</b>	<b>Verwendung</b>
SYSTEM.FTE.AU-THAGT1.agent_name	Berechtigung	Warteschlange zum Konfigurieren der Berechtigung zum Senden und Empfangen von Übertragungsanforderungen.
SYS-TEM.FTE.AUTHTRN1.agent_name	Berechtigung	Warteschlange zum Konfigurieren der Berechtigung zum Starten und Abbrechen verwalteter Übertragungen. Auch zum Starten verwalteter Aufrufe.
SYSTEM.FTE.AUTH-MON1.agent_name	Berechtigung	Warteschlange für die Konfiguration der Berechtigung, mit der ein Benutzer Ressourcenüberwachungen erstellen oder löschen kann, die von demselben Benutzer erstellt wurden.
SYSTEM.FTE.AU-THOPS1.agent_name	Berechtigung	Warteschlange zum Konfigurieren der Berechtigung zum Löschen von Ressourcenüberwachungen und geplanten Übertragungen, die von einem anderen Benutzer erstellt wurden.
SYS-TEM.FTE.AUTHSCH1.agent_name	Berechtigung	Warteschlange für die Konfiguration der Berechtigung zum Erstellen oder Löschen geplanter Übertragungen, die von demselben Benutzer erstellt wurden
SYSTEM.FTE.AU-THADM1.agent_name	Berechtigung	Warteschlange für die Konfiguration der Berechtigung zum Beenden des Agenten mit der Option <b>-m</b> im Befehl <b>fteStopAgent</b> .
SYSTEM.FTE.COM-MAND.agent_name	Operation	Warteschlange zum Senden von Befehlsanforderungen an einen Agenten.
SYSTEM.FTE.DATA.agent_name	Operation	Warteschlange, die von einem Zielagenten zum Halten von Daten verwendet wird, die von einem Quellenagenten gesendet wurden.

Tabelle 50. Warteschlangennamen, -typ und -verwendung (Forts.)

Warteschlangenname	Warteschlangentyp	Verwendung
SYSTEM.FTE.REPLY.agent_name	Operation	Warteschlange für den Empfang von Antworten von einem Zielagenten.
SYSTEM.FTE.STATE.agent_name	Operation	Warteschlange zum Halten des Status einer Übertragungsanforderung.
SYSTEM.FTE.EVENT.agent_name	Operation	Warteschlange für das Ressourcenüberwachungsprotokoll.

Wenn ein Agent an Übertragungen aus Nachrichten in Dateien oder aus Dateien in Nachrichten teilnimmt, muss möglicherweise die Definition der SYSTEM.FTE.STATE.agent\_name -Warteschlange geändert werden, damit diese verwalteten Übertragungen stattfinden können. Weitere Informationen hierzu finden Sie in [Anleitung zum Festlegen von MQ -Attributen und MFT-Eigenschaften, die der Nachrichtengröße zugeordnet sind](#).



**Achtung:** Sie sollten die Definitionen der anderen Systemwarteschlangen nicht ändern.

Außerdem darf das Thema SYSTEM.FTE nicht geändert oder gelöscht werden, da es auch nur zur internen Verwendung bestimmt ist.

### Temporäre Warteschlangen

Managed File Transfer erstellt zu mehreren Zwecken temporäre Warteschlangen. Der Name jeder Warteschlange beginnt mit WMQFTE. Standardmäßig. (Der Punkt ist Teil des Standardpräfixes.) Wenn Sie dieses Präfix ändern wollen, können Sie die Eigenschaft **dynamicQueuePrefix** in der Datei `command.properties` und/oder in der Datei `coordination.properties` verwenden. Mit der Eigenschaft in der Datei `command.properties` wird das Präfix der temporären Warteschlangen festgelegt, die für Antworten auf Befehle erstellt werden, bei denen eine Antwort vom Agenten erforderlich ist. Die Eigenschaft in der Datei `coordination.properties` wird verwendet, um das Präfix von temporären Warteschlangen festzulegen, die für andere Zwecke erstellt werden. Beispiel: `WMQFTE.FTE.TIMECHK.QUEUE`, wobei `WMQFTE` ist der Wert, der durch die Eigenschaft **dynamicQueuePrefix** definiert wird.

#### Zugehörige Verweise

[Benutzerberechtigungen für MFT-Agentenaktionen beschränken](#)

## Konventionen zum Benennen von MFT-Objekten

Verwenden Sie zum Benennen Ihrer Managed File Transfer-Objekte folgende Konventionen:

- Agenten- und Protokollnamen:
  - Dürfen maximal 28 Zeichen lang sein, wobei die Groß-/Kleinschreibung keine Rolle spielt.
  - Namen, die in Kleinschreibung oder in gemischter Groß-/Kleinschreibung eingegeben werden, werden in Großbuchstaben umgewandelt.
  - Müssen den Standardkonventionen zum Benennen von IBM MQ-Objekten entsprechen.

Diese Konventionen werden im Abschnitt [Regeln für die Benennung von IBM MQ-Objekten](#) näher beschrieben.
- Zusätzlich zu den IBM MQ-Konventionen für die Benennung von Objekten muss auch beachtet werden, dass:
  - in Agenten- und Protokollnamen keine Schrägstriche (/) verwendet werden dürfen.
  - in Agenten- und Protokollnamen keine Prozentzeichen (%) verwendet werden dürfen.
- Bei den Namen der Eigenschaften in den Eigenschaftendateien muss die Groß-/Kleinschreibung beachtet werden.

- Bei den Warteschlangenmanagernamen muss die Groß-/Kleinschreibung beachtet werden.
- Auf einigen Plattformen muss bei den Dateinamen die Groß-/Kleinschreibung beachtet werden.
- Ressourcenmonitor- und Übertragungsschablonennamen:
  - Groß-/Kleinschreibung muss nicht beachtet werden.
  - Namen, die in Kleinschreibung oder in gemischter Groß-/Kleinschreibung eingegeben werden, werden in Großbuchstaben umgewandelt.
  - Die Namen dürfen keinen Stern (\*), keine Prozentzeichen (%) oder keine Fragezeichen (?) enthalten.
- Protokolldateiservernamen müssen:
  - mindestens zwei Zeichen enthalten und sind in der Länge nicht begrenzt.
  - Groß-/Kleinschreibung muss nicht beachtet werden.
  - Müssen den Standardkonventionen zum Benennen von IBM MQ-Objekten entsprechen.

Diese Konventionen werden im Abschnitt [Regeln für die Benennung von IBM MQ-Objekten](#) näher beschrieben.

## Dateien im integrierten Dateisystem (Integrated File System, IFS) von IBM i

**IBM i**

Dateinamen im IFS dürfen keine der folgenden Zeichen enthalten:

- Umgekehrter Schrägstrich (\)
- Schrägstrich (/)
- Doppelpunkt (:)
- Stern (\*)
- Fragezeichen (?)
- Anführungszeichen (")
- Kleiner-als-Zeichen (<)
- Größer-als-Zeichen (>)
- Vertikaler Balken (|)

Wenn Sie versuchen, Dateien in das integrierte Dateisystem von IBM i zu übertragen, deren Namen diese Zeichen enthalten, schlägt die Übertragung fehl.

## Namen von Datasets

**z/OS**

Die Benennung von Datasets unterliegt einigen Beschränkungen, die sich auf die maximale Länge des Namens und die Zeichen beziehen, die für die Namen von Datasets verwendet werden können. Die Namen von Members partitionierter Dateien dürfen maximal acht Zeichen umfassen, das Punktzeichen (.) ist nicht zulässig. Bei der Übertragung in ein Dataset muss der Name explizit angegeben werden; diese Namenseinschränkungen stellen also kein Problem dar. Wenn Sie jedoch eine Übertragung aus Dateien in Mitglieder der partitionierten Datei vornehmen, wird der Dateipfad möglicherweise keinem Mitgliedsnamen der partitionierten Datei zugeordnet. Bei einer Übertragung in eine partitionierte Datei (PDS-Dataset) wird jede Quelldatei zu einem Member der partitionierten Datei und die Namen der einzelnen Members werden auf der Basis des Quellennamens generiert.

Bei den Membernamen der partitionierten Datei handelt es sich um nicht qualifizierte z/OS-Namen, die durch den folgenden regulären Ausdruck definiert werden:

```
[a-zA-Z$#@] [a-zA-Z0-9$#@]{0-7}
```

Mit dem folgenden Schema wird ein Quellendataset oder ein Quellendateiname in einen gültigen Mitgliedsnamen der partitionierten Datei konvertiert. Hierbei gelten diese Aspekte in der folgenden Reihenfolge:

1. Es werden nur die Zeichen im Namen verwendet, die auf den letzten Schrägstrich (/), den letzten Backslash (\) oder den letzten Doppelpunkt (:) folgen. Es wird also nur der Namensbereich eines Dateipaths verwendet.
2. Bei Quellendateien (nicht bei Datasets oder Members partitionierter Dateien) werden alle Zeichen ab dem letzten Punktzeichen (.) ignoriert, und zwar einschließlich des Punkts.
3. Umfassen Namen mehr als acht Zeichen, werden nur die letzten acht Zeichen verwendet.
4. Punkte werden durch kommerzielle A-Zeichen (@) ersetzt.
5. Ungültige Zeichen werden durch kommerzielle A-Zeichen (@) ersetzt.
6. Wenn die Konvertierung keine Zeichen ergibt, lautet das Member der partitionierten Datei @.

## V 9.1.4 Statusnachrichten von MFT-Agenten

Hochverfügbarkeitsagenten veröffentlichen Statusinformationen im XML-Format.

### Beispiel-XML mit Informationen zu drei Standby-Instanzen

```
<?xml version="1.0" encoding="UTF-8"?>
<AgentStandbyStatus version="6.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" xsi:noNamespa
ceSchemaLocation="AgentStandbyStatus.xsd">
  <instance host="9.122.123.124" agentVersion="9.1.4.0" />
  <instance host="agenthost.ibm.com" agentVersion="9.1.4.0" />
  <instance host="10.11.12.14" agentVersion="9.1.4.0" />
</AgentStandby>
```

### Veröffentlichung des Agentenstatus mit integrierter Standby-Status-XML

Die Standby-Status-XML wird in Fettdruck angezeigt.

```
<?xml version="1.0" encoding="UTF-8"?>
<properties version="1.0">
  <entry key="SourceTransferStates"/>
  <entry key="queueManagerPort">1414</entry>
  <entry key="agentStandbyInstances">&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;AgentStandbyStatus
version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="AgentStandbyStatus.xsd"&gt;&lt;Instances&gt;&lt;instance
host="9.122.123.124"
  agentVersion="9.1.4.0" /&gt;&lt;instance host="agenthost.ibm.com" agentVersi
on="9.1.4.0" /&gt;&lt;instance host="10.11.12.14"
  agentVersion="9.1.4.0" /&gt;&lt;/Instances&gt;&lt;/AgentStandbyStatus&gt;</entry>
  <entry key="agentType">STANDARD</entry>
  <entry key="agentDeclaredHostName">MFTHA1</entry>
  <entry key="agentDescription"/>
  <entry key="maxQueuedTransfers">1000</entry>
  <entry key="agentTimeZone">America/Los_Angeles</entry>
  <entry key="agentOsName">Windows Server 2012 R2</entry>
  <entry key="PublishTimeUTC">2019-05-22T06:02:50Z</entry>
  <entry key="queueManagerHost">localhost</entry>
  <entry key="AgentStartTimeUTC">2019-05-22T04:13:02Z</entry>
  <entry key="agentTraceLevel">&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;
agentTraceStatus version="6.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="AgentTraceStatus.xsd"&gt;&lt;trace
level="all"&gt;com.ibm.wmqfte&lt;/trace&gt;&lt;/agentTraceStatus&gt;</entry>
  <entry key="DestinationTransferStates"/>
  <entry key="queueManager">MFTHAQM</entry>
  <entry key="agentProductVersion">9.1.4.0</entry>
  <entry key="AgentStatusPublishRate">300</entry>
  <entry key="maxSourceTransfers">25</entry>
  <entry key="AgentStatus">STARTED</entry>
  <entry key="maxDestinationTransfers">25</entry>
  <entry key="agentName">SRC</entry>
  <entry key="CommandTimeUTC">2019-05-22T06:02:50Z</entry>
  <entry key="queueManagerChannel">MFT_HA_CHN</entry>
  <entry key="agentInterfaceVersion">6.00</entry>
  <entry key="agentVersion">p914-L191119</entry>
</properties>
```

## Zugehörige Verweise

[fteCreateAgent](#)

[Agent, GET](#)

# IBM MQ Internet Pass-Thru -Konfigurationsreferenz

---

IBM MQ Internet Pass-Thru (MQIPT) verwendet eine Konfigurationsdatei namens `mqipt.conf`, um Routen zu definieren und die Aktionen des MQIPT-Servers zu steuern.

Die Konfigurationsdatei enthält eine Reihe von Abschnitten. Es gibt einen Abschnitt `[global]` und einen zusätzlichen Abschnitt `[route]` für jede Route, die über MQIPT definiert wurde.

Jeder Abschnitt enthält Name/Wert-Eigenschaftspaare. Einige Eigenschaften können nur im Abschnitt `[global]` angezeigt werden, einige können nur in den `[route]`-Abschnitten angezeigt werden, und einige können sowohl im Abschnitt `[route]` als auch in `[global]` angezeigt werden. Wenn eine Eigenschaft sowohl im Abschnitt "route" als auch im Abschnitt `[global]` angezeigt wird, überschreibt der Wert der Eigenschaft im Abschnitt `[route]` den globalen Wert, aber nur für die betreffende Route. Auf diese Weise kann der Abschnitt `[global]` verwendet werden, um die Standardwerte festzulegen, die für die Eigenschaften verwendet werden sollen, die nicht in den einzelnen `[route]`-Abschnitten festgelegt sind.

Der Abschnitt `[global]` beginnt mit einer Zeile, in der die Zeichenfolge `[global]` steht, und endet am Beginn des ersten `[route]`-Abschnitts. Der Abschnitt `[global]` muss in der Datei vor allen Abschnitten `[route]` angegeben sein.

Jeder `[route]`-Abschnitt beginnt mit einer Zeile, die die Zeichenfolge `[route]` enthält, und endet, wenn der nächste `[route]`-Abschnitt beginnt oder wenn das Ende der Konfigurationsdatei erreicht ist.

Nicht erkannte Eigenschaftsnamen werden ignoriert. Wenn eine Eigenschaft in einem `[route]`-Abschnitt einen anerkannten Namen, aber keinen gültigen Wert hat (z. B. `MinConnectionThreads=x` oder `HTTP=unsure`), ist diese Route inaktiviert (d. h. sie ist nicht empfangsbereit für eingehenden Verbindungen). Wenn eine Eigenschaft im `[global]`-Abschnitt einen anerkannten Namen, aber keinen gültigen Wert hat, sind alle Routen inaktiviert und MQIPT wird nicht gestartet. Wenn für eine Eigenschaft `true` oder `false` als mögliche Werte aufgelistet sind, kann eine beliebige Mischung aus Groß- und Kleinschreibung verwendet werden.



**Achtung:** Der Maximalwert für die Anzahl der Routen, die in der Datei `mqipt.conf` hinzugefügt werden können, ist 100.

Sie können den Wert einer Eigenschaft ändern, indem Sie entweder die Datei `mqipt.conf` bearbeiten oder indem Sie die IPT Administration Client-GUI verwenden. Wenn Sie Änderungen übernehmen wollen, aktualisieren Sie MQIPT, entweder über die IPT Administration Client-GUI oder über den Befehl `mqiptAdmin -refresh`.

Um Kommentare in die Konfigurationsdatei aufzunehmen, beginnen Sie eine Zeile mit dem Zeichen "#".

Änderungen an bestimmten Eigenschaften bewirken nur dann den Neustart einer Route, wenn andere Eigenschaften bereits aktiviert sind. Alle Änderungen an den HTTP-Eigenschaften haben z. B. nur einen Effekt, wenn die Eigenschaft **HTTP** auch aktiviert ist.

Wenn eine Route erneut gestartet wird, werden bestehende Verbindungen beendet. Um dieses Verhalten außer Kraft zu setzen, legen Sie für die Eigenschaft **RouteRestart** den Wert `false` fest. Dies verhindert den Neustart der Route, sodass vorhandene Verbindungen aktiv bleiben können, bis die Eigenschaft **RouteRestart** erneut aktiviert wird.

Informationen zum Einrichten einiger einfacher Konfigurationen finden Sie unter [Erste Schritte mit MQIPT](#). Eine Beispielkonfiguration finden Sie in der Datei `mqiptSample.conf` im MQIPT-Installationsverzeichnis.

## Zusammenfassung der MQIPT-Eigenschaften

Diese Tabelle zeigt eine Zusammenfassung der MQIPT-Konfigurationseigenschaften und enthält folgende Informationen:



- Eine alphabetische Liste der MQIPT-Eigenschaften mit Links zu weiteren Informationen im Abschnitt [route] oder im Abschnitt [global], wenn der Abschnitt [route] nicht zutrifft.
- Die Eigenschaft, die für einen Wert auf true gesetzt werden muss, damit ein Wert wirksam wird.
- Gibt an, ob die Eigenschaft für den Abschnitt [global] und/oder den Abschnitt [route] gilt.
- Standardwerte, die verwendet werden, wenn eine Eigenschaft sowohl im Abschnitt [route] als auch im Abschnitt [global] fehlt. Beim Angeben der Werte true und false kann eine beliebige Mischung aus Groß- und Kleinbuchstaben verwendet werden.



Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
<a href="#">AccessPW</a>		ja	nein	null
<a href="#">Aktiv</a>		ja	ja	true
<a href="#">ClientAccess</a>		ja	ja	false
<a href="#">CommandPort</a>		ja	nein	null
<a href="#">ConnectionLog</a>		ja	nein	true
<a href="#">Destination</a>		nein	ja	null
<a href="#">DestinationPort</a>		nein	ja	1414
<a href="#">„[V9.1.4 Dezember 2019]EnableAdvancedCapabilities“ auf Seite 263</a>		ja	nein	false
<a href="#">HTTP</a>		ja	ja	false
<a href="#">HTTPProxy</a>	HTTP	ja	ja	null
<a href="#">HTTPProxyPort</a>	HTTP	ja	ja	8080
<a href="#">HTTPS</a>	HTTP	ja	ja	false
<a href="#">HTTPTServer</a>	HTTP	ja	ja	null
<a href="#">HTTPTServerPort</a>	HTTP	ja	ja	null
<a href="#">IdleTimeout</a>		ja	ja	0
<a href="#">IgnoreExpiredCRLs</a>		ja	ja	false
<a href="#">LDAP</a>		ja	ja	false
<a href="#">LDAPIgnoreErrors</a>	LDAP	ja	ja	false
<a href="#">LDAPCacheTimeout</a>	LDAP	ja	ja	24
<a href="#">LDAPServer1</a>	LDAP	ja	ja	null
<a href="#">LDAPServer1Port</a>	LDAP	ja	ja	389
<a href="#">LDAPServer1Userid</a>	LDAP	ja	ja	null
<a href="#">LDAPServer1Password</a>	LDAP	ja	ja	null
<a href="#">LDAPServer1Timeout</a>	LDAP	ja	ja	0
<a href="#">LDAPServer2</a>	LDAP	ja	ja	null
<a href="#">LDAPServer2Port</a>	LDAP	ja	ja	389
<a href="#">LDAPServer2Userid</a>	LDAP	ja	ja	null
<a href="#">LDAPServer2Password</a>	LDAP	ja	ja	null

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
<a href="#">LDAPServer2Timeout</a>	LDAP	ja	ja	0
<a href="#">ListenerAddress</a>		ja	ja	null
<a href="#">ListenerPort</a>		nein	ja	null
<a href="#">LocalAddress</a>		ja	ja	null
<a href="#">MaxConnectionThreads</a>		ja	ja	100
<a href="#">MaxLogFileSize</a>		ja	nein	50
<a href="#">MinConnectionThreads</a>		ja	ja	5
<a href="#">Name</a>		nein	ja	null
<a href="#">OutgoingPort</a>		nein	ja	0
<a href="#">QMgrAccess</a>		ja	ja	true
<a href="#">RemoteShutdown</a>		ja	nein	false
<a href="#">RouteRestart</a>		ja	ja	true
<a href="#">SecurityExit</a>		ja	ja	false
<a href="#">SecurityExitName</a>	SecurityExit	ja	ja	null
<a href="#">SecurityExitPath</a>	SecurityExit	ja	ja	<i>mqipt_home\exits</i>
<a href="#">SecurityExitTimeout</a>	SecurityExit	ja	ja	30
<a href="#">SecurityManager</a>		ja	nein	false
<a href="#">SecurityManagerPolicy</a>		ja	nein	null
<a href="#">SocksClient</a>		ja	ja	false
<a href="#">SocksProxyHost</a>	SocksClient	ja	ja	null
<a href="#">SocksProxyPort</a>	SocksClient	ja	ja	1080
<a href="#">SocksServer</a>		ja	ja	false
<a href="#">SSLClient</a>		ja	ja	false
<a href="#">SSLClientCAKeyRing</a>	SSLClient	ja	ja	null
<a href="#">SSLClientCAKeyRingPW</a>	SSLClient	ja	ja	null
<a href="#">„[V9.1.4 Dezember 2019]SSLClientCAKeyRingUseCryptoHardware“ auf Seite 270</a>	SSLClient	ja	ja	false
<a href="#">SSLClientCipherSuites</a>	SSLClient	ja	ja	null
<a href="#">SSLClientConnectTimeout</a>	SSLClient	ja	ja	30
<a href="#">SSLClientDN_C</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_CN</a>	SSLClient	ja	ja	* (Anmerkung 1)

<b>Eigenschaftsname</b>	<b>Auf true zu setzende Eigenschaft</b>	<b>Global</b>	<b>Route</b>	<b>Standard</b>
<a href="#">SSLClientDN_DC</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_DNQ</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_L</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_O</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_OU</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_PC</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_ST</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_Street</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_T</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientDN_UID</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientExit</a>		ja	ja	false
<a href="#">SSLClientKeyRing</a>	SSLClient	ja	ja	null
<a href="#">SSLClientKeyRingPW</a>	SSLClient	ja	ja	null
<a href="#">„[V9.1.4 Dezember 2019]SSLClient-KeyRingUseCryptoHardware“ auf Seite 272</a>	SSLClient	ja	ja	false
<a href="#">SSLClientProtocols</a>	SSLClient	ja	ja	TLSv1.2
<a href="#">SSLClientSiteDN_C</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_CN</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_DC</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_DNQ</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_L</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_O</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_OU</a>	SSLClient	ja	ja	* (Anmerkung 1)

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
<a href="#">SSLClientSiteDN_PC</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_ST</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_Street</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_T</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteDN_UID</a>	SSLClient	ja	ja	* (Anmerkung 1)
<a href="#">SSLClientSiteLabel</a>	SSLClient	ja	ja	null
<a href="#">SSLExitData</a>	SSLServerExit	ja	ja	null
<a href="#">SSLExitName</a>	SSLServerExit	ja	ja	null
<a href="#">SSLExitPath</a>	SSLServerExit	ja	ja	<i>mqipt_home\exits</i>
<a href="#">SSLExitTimeout</a>	SSLServerExit	ja	ja	30
<a href="#">SSLProxyMode</a>		ja	ja	false
<a href="#">SSLPlainConnections</a>	entweder SSLServer oder SSLProxyMode	ja	ja	false
<a href="#">SSLServer</a>		ja	ja	false
<a href="#">SSLServerAskClientAuh</a>	SSLServer	ja	ja	false
<a href="#">SSLServerCAKeyRing</a>	SSLServer	ja	ja	null
<a href="#">SSLServerCAKeyRingPW</a>	SSLServer	ja	ja	null
<a href="#">„[V9.1.4 Dezember 2019]SSLServerCAKeyRingUseCryptoHardware“ auf Seite 276</a>	SSLServer	ja	ja	false
<a href="#">SSLServerCipherSuites</a>	SSLServer	ja	ja	null
<a href="#">SSLServerDN_C</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_CN</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_DC</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_DNQ</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_L</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_O</a>	SSLServer	ja	ja	* (Anmerkung 1)

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
<a href="#">SSLServerDN_OU</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_PC</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_ST</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_Street</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_T</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerDN_UID</a>	SSLServer	ja	ja	* (Anmerkung 1)
SSLServerExit		ja	ja	false
SSLServerKeyRing	SSLServer	ja	ja	null
SSLServerKeyRingPW	SSLServer	ja	ja	null
„[V9.1.4 Dezember 2019]SSLServerKeyRingUseCryptoHardware“ auf Seite 279	SSLServer	ja	ja	false
<a href="#">SSLServerProtocols</a>	SSLServer	ja	ja	TLSv1.2
<a href="#">SSLServerSiteDN_C</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_CN</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_DC</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_DNQ</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_L</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_O</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_OU</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_PC</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_ST</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_Street</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteDN_T</a>	SSLServer	ja	ja	* (Anmerkung 1)

Eigenschaftsname	Auf true zu setzende Eigenschaft	Global	Route	Standard
<a href="#">SSLServerSiteDN_UID</a>	SSLServer	ja	ja	* (Anmerkung 1)
<a href="#">SSLServerSiteLabel</a>	SSLServer	ja	ja	null
  <a href="#">StoredCredentialsFormat</a>		ja	ja	null
<a href="#">TCPKeepAlive</a>		ja	ja	false
<a href="#">Trace</a>		ja	ja	0
<a href="#">UriName</a>	HTTP	ja	ja	(Anmerkung 2)

#### Anmerkungen:

1. Der Stern (\*) steht für ein Platzhalterzeichen.
2. Ausführliche Informationen zu den Standardeinstellungen finden Sie unter [UriName](#) im Abschnitt „MQIPT-Routeneigenschaften“ auf Seite 264.

#### Zugehörige Verweise

„IBM MQ Internet Pass-Thru -Konfigurationsreferenz“ auf Seite 256

IBM MQ Internet Pass-Thru (MQIPT) verwendet eine Konfigurationsdatei namens `mqipt.conf`, um Routen zu definieren und die Aktionen des MQIPT-Servers zu steuern.

„Globale MQIPT-Eigenschaften“ auf Seite 262

Die Konfigurationsdatei `mqipt.conf` kann eine Reihe globaler Eigenschaften enthalten.

„MQIPT-Routeneigenschaften“ auf Seite 264

Die Konfigurationsdatei `mqipt.conf` kann Eigenschaften für einzelne Routen enthalten.


## Globale MQIPT-Eigenschaften

Die Konfigurationsdatei `mqipt.conf` kann eine Reihe globaler Eigenschaften enthalten.

Die folgenden Eigenschaften können nur im Abschnitt `[global]` von `mqipt.conf` angezeigt werden. Alle Routeneigenschaften außer **ListenerPort**, **Destination**, **DestinationPort**, **Name** und **OutgoingPort** können auch im Abschnitt `[global]` angezeigt werden. Wenn eine Eigenschaft sowohl im Abschnitt `"route"` als auch im Abschnitt `[global]` angezeigt wird, überschreibt der Wert der Eigenschaft im Abschnitt `[route]` den globalen Wert, aber nur für die betreffende Route. Auf diese Weise kann der Abschnitt `[global]` verwendet werden, um die Standardwerte festzulegen, die für die Eigenschaften verwendet werden sollen, die nicht in den einzelnen `[route]` -Abschnitten festgelegt sind.

#### AccessPW

Das Kennwort, das verwendet wird, wenn ein IPT Administration Client Befehle an MQIPT sendet. Wenn diese Eigenschaft nicht vorhanden oder leer ist, findet keine Überprüfung statt.

 Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl `mqiptPW` verschlüsselt wurde, oder um ein Klartextkennwort. Klartextkennwörtern können nur alphanumerische Zeichen enthalten. Es wird dringend empfohlen, die in der MQIPT-Konfiguration gespeicherten Kennwörter zu verschlüsseln. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

#### CommandPort

Der TCP/IP-Port, auf dem der MQIPT für Konfigurationsbefehle vom Dienstprogramm `mqiptAdmin` oder vom IPT Administration Client empfangsbereit ist. Sie können den Befehlsports im IPT Administration Client auf die gleiche Weise wie jede andere Eigenschaft ändern. Sie ändern dadurch nicht

die Verbindungseigenschaften. Wenn Sie die neue Einstellung in MQIPT anwenden, ändert der IPT Administration Client die Verbindungseigenschaften automatisch.

Wenn die Eigenschaft **CommandPort** nicht vorhanden ist, ist MQIPT nicht für Konfigurationsbefehle empfängsbereit. Wenn Sie die Standardportnummer 1881 verwenden möchten, die standardmäßig vom IPT Administration Client und vom Script **mqiptAdmin** über die Befehlszeile verwendet wird, setzen Sie **CommandPort** auf 1881. Dieser Wert wird für Sie festgelegt, wenn Sie die Konfigurationsdatei `mqiptSample.conf` verwenden.

### ConnectionLog

Zulässig sind die Werte `true` oder `false`. Bei Angabe von 'true' protokolliert MQIPT alle (erfolgreichen oder erfolglosen) Verbindungsversuche im Unterverzeichnis `logs` sowie Ereignisse für Verbindungsbeendigungen in der Datei `mqiptYYYYMMDDHHmmSS.log` (die Zeichen `YYYYMMDDHHmmSS` geben dabei das aktuelle Datum und die aktuelle Uhrzeit an). Der Standardwert von **ConnectionLog** ist `Wahr`. Wenn diese Eigenschaft von `true` in `false` geändert wird, schließt MQIPT das vorhandene Verbindungsprotokoll und erstellt ein neues. Das neue Protokoll wird verwendet, wenn die Eigenschaft auf `true` zurückgesetzt wird.

### V 9.1.4 EnableAdvancedCapabilities

Setzen Sie diese Eigenschaft auf `true`, um zu bestätigen, dass erweiterte Funktionen, die eine IBM MQ Advanced-, IBM MQ Appliance- oder IBM MQ Advanced for z/OS VUE-Berechtigung erfordern, von MQIPT verwendet werden können. Mit der entsprechenden Berechtigung können Sie die erweiterten Funktionen in MQIPT verwenden. Wenn erweiterte Funktionen auf einer Route aktiviert sind, muss auch der lokale Warteschlangenmanager, der über die MQIPT-Route verbunden ist, über eine IBM MQ Advanced-, IBM MQ Appliance- oder IBM MQ Advanced for z/OS VUE-Berechtigung verfügen. Routen, die erweiterte Funktionen verwenden, können nur gestartet werden, wenn diese Eigenschaft auf `true` gesetzt ist. Wenn diese Eigenschaft von `true` in `false` geändert wird, werden Routen, die erweiterte Funktionen verwenden, gestoppt.

### MaxLogFileSize

Die maximale Größe der Verbindungsprotokolldatei (in KB). Wenn die Größe dieser Datei den maximalen Wert überschreitet, wird eine Sicherungskopie (`mqipt001.log`) erstellt und eine neue Datei gestartet. Es werden nur zwei Sicherungsdateien gespeichert (`mqipt001.log` und `mqipt002.log`); frühere Sicherungsdateien werden gelöscht, sobald die Hauptprotokolldatei voll ist. Der Standardwert von **MaxLogFileSize** ist 50; der zulässige Mindestwert ist 5.

### RemoteShutDown

Zulässig sind die Werte `true` oder `false`. Wenn 'true' angegeben ist (und ein Befehlsport vorhanden ist), wird MQIPT beendet, sobald im Befehlsport ein Stoppbefehl empfangen wird. Der Standardwert ist `false`.

### SecurityManager

Setzen Sie diese Eigenschaft auf `true`, um den Java security manager für diese Instanz von MQIPT zu aktivieren. Sie müssen sicherstellen, dass die korrekten Berechtigungen erteilt sind. Weitere Informationen finden Sie in [Java security manager](#). Der Standardwert für diese Eigenschaft ist `false`.

### SecurityManagerPolicy

Der vollständig qualifizierte Dateiname einer Richtliniendatei. Wenn diese Eigenschaft nicht festgelegt ist, werden nur die Dateien für das Standardsystem und die Benutzerrichtlinien verwendet. Wenn der Java security manager bereits aktiviert ist, haben Änderungen an dieser Eigenschaft erst Auswirkungen, wenn der Java security manager inaktiviert und erneut aktiviert wurde.

### Trace

Die Tracestufe für globale MQIPT-Threads, die keiner Route zugeordnet sind, und für Routen, für die die Eigenschaft **Trace** nicht festgelegt ist. Beispielsweise werden der Hauptsteuerungsthread von MQIPT und der Listener-Thread für den Befehlsport keiner Route zugeordnet und werden nur aufgezeichnet, wenn die Tracefunktion im Abschnitt `[global]` aktiviert ist. Der Wert der Eigenschaft **Trace** in einem Abschnitt `[route]` überschreibt die globale Eigenschaft **Trace** für diese Route. Informationen zum Tracing von Threads, die einer Route zugeordnet sind, finden Sie unter [Trace](#) im Abschnitt `[route]`.

Diese Eigenschaft kann einen der folgenden Werte haben:

**0**

Trace ist nicht aktiviert

**Jede beliebige positive Ganzzahl.**

Trace ist aktiviert

Der Standardwert ist 0.

## MQIPT-Routeneigenschaften

Die Konfigurationsdatei `mqipt.conf` kann Eigenschaften für einzelne Routen enthalten.

Der Abschnitt `[route]` der Konfigurationsdatei `mqipt.conf` kann die folgenden Eigenschaften enthalten:

### Aktiv

Die Route akzeptiert eingehende Verbindungen nur, wenn der Wert von **Active** auf `Wahr` gesetzt ist. Dies bedeutet, dass Sie den Zugriff auf das Ziel vorübergehend inaktivieren können, indem Sie diesen Wert auf `Falsch` setzen, ohne den Abschnitt `[route]` aus der Konfigurationsdatei löschen zu müssen. Wenn Sie den Wert dieser Eigenschaft in `false` ändern, wird die Route gestoppt, wenn ein Befehl zur Aktualisierung ausgegeben wird. Alle Verbindungen zu der Route werden gestoppt.

### ClientAccess

Die Route lässt eingehende Clientkanalverbindungen nur zu, wenn der Wert von **ClientAccess** auf `Wahr` gesetzt ist. Beachten Sie, dass Sie MQIPT auch so konfigurieren können, dass nur Clientanforderungen, nur Warteschlangenmanageranforderungen oder beide Anforderungstypen akzeptiert werden. Verwenden Sie diese Eigenschaft zusammen mit der Eigenschaft **QMGrAccess**. Wenn Sie diese Eigenschaft in `false` ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

### Ziel

Der Hostname (oder die IP-Adresse in Schreibweise mit Trennzeichen) des Warteschlangenmanagers, oder der nachfolgenden Instanz von MQIPT, zu der diese Route eine Verbindung herstellen soll. Jeder Abschnitt `[route]` muss einen expliziten **Destination**-Wert enthalten, aber mehrere Abschnitte `[route]` können auf dasselbe Ziel verweisen. Wenn sich eine Änderung dieser Eigenschaft auf die Route auswirkt, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt. Bei Verwendung der Eigenschaft **SocksProxyHost** muss die Eigenschaft **Destination** das IPv4-Adressformat in Schreibweise mit Trennzeichen verwenden.

### DestinationPort

Der Port auf dem Zielhost, zu dem diese Route eine Verbindung herstellen soll. Jeder `[route]`-Abschnitt muss einen expliziten **DestinationPort**-Wert enthalten, aber mehrere Routen können auf dieselbe Kombination von **Destination**- und **DestinationPort**-Werten verweisen. Wenn sich eine Änderung dieser Eigenschaft auf die Route auswirkt, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

### HTTP

Setzen Sie **HTTP** für Routen, die für abgehende HTTP-Tunnelungsanforderungen zuständig sind, auf `Wahr`. Die Eigenschaft **Destination** für die Route muss der Hostname eines anderen MQIPT sein, wenn **HTTP** auf `wahr` gesetzt ist. Setzen Sie **HTTP** für mit IBM MQ-Warteschlangenmanagern verbundene Routen auf `Falsch`. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt. Mindestens eine der Eigenschaften **HTTPProxy** oder **HTTPServer** muss ebenfalls angegeben werden, wenn **HTTP** auf `wahr` gesetzt ist. Diese Eigenschaft kann nicht zusammen mit der Eigenschaft **SocksClient** verwendet werden.

### HTTPProxy

Der Hostname (oder die IP-Adresse in Schreibweise mit Trennzeichen) des HTTP-Proxys, der von allen Verbindungen für diese Route verwendet wird. Eine **CONNECT**-Anforderung wird an den HTTP-Proxy ausgegeben, anstelle der **POST**-Anforderung, die normalerweise verwendet wird, wenn kein HTTP-Proxy konfiguriert ist. Wenn Sie diese Eigenschaft ändern (und **HTTP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.



## HTTPProxyPort

Die Portadresse, die im HTTP-Proxy verwendet werden soll. Der Standardwert ist 8080. Wenn Sie diese Eigenschaft ändern (und **HTTTP** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

## HTTPServer

Der Hostname (oder die IP-Adresse in Schreibweise mit Trennzeichen) des HTTP-Servers, der von allen Verbindungen für diese Route verwendet wird. Dies ist in der Regel der Hostname eines anderen MQIPT.

Wenn **HTTTPProxy** nicht angegeben ist, stellt MQIPT eine Verbindung zu dem in **HTTPServer** angegebenen Host her und gibt HTTP-Anforderungen **POST** an den Host aus, der in der Eigenschaft **Destination** der Route angegeben ist. Wenn **HTTTPProxy** angegeben ist, stellt MQIPT stattdessen eine Verbindung zu dem in **HTTTPProxy** angegebenen Host her und fordert an, dass der Proxy einen Tunnel zu dem in **HTTPServer** angegebenen Host herstellt.

Wenn **HTTTPProxy** angegeben wird, ist der Standardwert die Route **Destination**.

Wenn Sie diese Eigenschaft ändern (und **HTTTP** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

## HTTPS

Setzen Sie **HTTTPS** auf **Wahr**, wenn HTTPS-Anforderungen abgesetzt werden sollen. Die Eigenschaften **HTTTP** und **SSLClient** müssen ebenfalls aktiviert sein und der Clientschlüsselring muss wie bei einer SSL/TLS-Operation mit der Eigenschaft **SSLClientKeyRing** oder **SSLClientKeyRingUseCryptoHardware** konfiguriert werden. Wenn Sie die Eigenschaft **HTTTPS** ändern (und **HTTTP** auf **Wahr** gesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

## HTTPServerPort

Die Portadresse, die im HTTP-Server verwendet werden soll. Der Standardwert ist 8080, sofern **HTTTPProxy** nicht angegeben ist. In diesem Fall ist der Standardwert die Route **DestinationPort**.

Wenn Sie diese Eigenschaft ändern (und **HTTTP** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

## IdleTimeout

Die Zeit in Minuten, nach der eine inaktive Verbindung geschlossen wird. Beachten Sie, dass Warteschlangenmanager-zu-Warteschlangenmanager-Kanäle auch über die Eigenschaft **DISCINT** verfügen. Wenn Sie den Parameter **IdleTimeout** festlegen, beachten Sie **DISCINT**. Wenn **IdleTimeout** auf 0 gesetzt ist, gibt es kein Inaktivitätszeitlimit. Änderungen an dieser Eigenschaft werden nur bei einem Neustart der Route wirksam.

## IgnoreExpiredCRLs

Setzen Sie **IgnoreExpiredCRLs** auf **Wahr**, um eine abgelaufene CRL zu ignorieren. Der Standardwert ist **false**. Wenn Sie **IgnoreExpiredCRLs** auf **Wahr** setzen, könnte ein widerrufenes Zertifikat verwendet werden, um eine SSL/TLS-Verbindung herzustellen.

## LDAP

Setzen Sie **LDAP** auf **Wahr**, um die Verwendung eines LDAP-Servers bei Verwendung von SSL/TLS-Verbindungen zu aktivieren. MQIPT nutzt den LDAP-Server, um CRLs und ARLs abzurufen. Die Eigenschaft **SSLClient** oder **SSLServer** muss ebenfalls auf **Wahr** gesetzt sein, damit diese Eigenschaft wirksam wird.

## LDAPCacheTimeout

Die Ablaufzeit des temporären Cache in Stunden, die eine von einem LDAP-Server abgerufene CRL gespeichert wird. Danach wird der gesamte CRL-Cache geleert. Wenn Sie beispielsweise einen Wert von 1 Stunde angeben, wird der Cache einmal pro Stunde geleert. Der Standardwert ist 24. Wenn Sie einen Zeitlimitwert von 0 angeben, verfallen die Einträge im Cache erst, wenn die Route erneut gestartet wird. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPIgnoreErrors**

Setzen Sie **LDAPIgnoreErrors** auf `Wahr`, wenn beim Ausführen einer LDAP-Suche alle Verbindungs- oder Zeitlimitfehler ignoriert werden sollen. Wenn MQIPT keine erfolgreiche Suche durchführen kann, kann die Clientverbindung nur dann vollständig abgeschlossen werden, wenn diese Eigenschaft aktiviert wurde. Bei einer erfolgreichen Suche wurde eine CRL abgerufen oder es sind keine CRLs für die angegebene CA verfügbar. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

**Anmerkung:** Wenn Sie diese Eigenschaft aktivieren, kann eine SSL/TLS-Verbindung mithilfe eines widerrufenen Zertifikats hergestellt werden.

### **LDAPServer1**

Der Hostname oder die IP-Adresse des LDAP-Hauptserver. Falls LDAP auf `true` gesetzt ist, muss diese Eigenschaft festgelegt sein. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPServer1Port**

Die Nummer des Empfangsports auf dem LDAP-Hauptserver. Der Standardwert ist 389. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPServer1Userid**

Die Benutzer-ID, die für den Zugriff auf den LDAP-Hauptserver erforderlich ist. Diese Eigenschaft muss festgelegt sein, wenn eine Autorisierung für den Zugriff auf den LDAP-Hauptserver erforderlich ist. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPServer1Password**

Das Kennwort, das für den Zugriff auf den LDAP-Hauptserver erforderlich ist. Diese Eigenschaft muss festgelegt werden, wenn **LDAPServer1Userid** auf `Wahr` gesetzt wurde. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

**V 9.1.5** Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um ein Klartextkennwort. Klartextkennwörter können nur alphanumerische Zeichen enthalten. Es wird dringend empfohlen, die in der MQIPT-Konfiguration gespeicherten Kennwörter zu verschlüsseln. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

### **LDAPServer1Timeout**

Die Zeit in Sekunden, die MQIPT auf eine Antwort vom LDAP-Hauptserver wartet. Der Standardwert ist 0, was bedeutet, dass das Zeitlimit der Verbindung nie überschritten wird. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPServer2**

Der Hostname oder die IP-Adresse des LDAP-Sicherungsservers. Diese Eigenschaft ist optional. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPServer2Port**

Die Nummer des Empfangsports für den LDAP-Sicherungsserver. Der Standardwert ist 389. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPServer2Userid**

Die Benutzer-ID, die für den Zugriff auf den LDAP-Sicherungssever erforderlich ist. Diese Eigenschaft muss festgelegt sein, wenn eine Autorisierung für den Zugriff auf den LDAP-Sicherungssever erforderlich ist. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **LDAPServer2Password**

Das Kennwort, das für den Zugriff auf den LDAP-Sicherungssever erforderlich ist. Diese Eigenschaft muss festgelegt werden, wenn **LDAPServer2** auf `Wahr` gesetzt wurde. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **V 9.1.5**

Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl `mqiptPW` verschlüsselt wurde, oder um ein Klartextkennwort. Klartextkennwörtern können nur alphanumerische Zeichen enthalten. Es wird dringend empfohlen, die in der MQIPT-Konfiguration gespeicherten Kennwörter zu verschlüsseln. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

### **LDAPServer2Timeout**

Die Zeit in Sekunden, die MQIPT auf eine Antwort vom LDAP-Sicherungssever wartet. Der Standardwert ist 0, was bedeutet, dass das Zeitlimit der Verbindung nie überschritten wird. Wenn Sie diese Eigenschaft ändern (und **LDAP** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **ListenerAddress**

Verwenden Sie diese Eigenschaft, wenn das MQIPT-System über mehrere IP-Adressen verfügt und Sie den Listener-Port für die Route an eine bestimmte Adresse binden müssen. Dies ist hilfreich, wenn Sie eingehende Verbindungen auf die aus einer bestimmten Netzchnittstelle beschränken möchten. Der Wert dieser Eigenschaft sollte eine IP-Adresse für eine der Netzchnittstellen auf dem System sein, auf dem MQIPT ausgeführt wird. Standardmäßig werden Verbindungen aus allen Netzchnittstellen akzeptiert.

### **ListenerPort**

Die Portnummer, an der die Route für eingehende Anforderungen empfangsbereit sein sollte. Jeder `[route]`-Abschnitt muss einen expliziten **ListenerPort**-Wert enthalten. Die in jedem Abschnitt festgelegten **ListenerPort**-Werte müssen eindeutig sein. Es kann jede gültige Portnummer verwendet werden, einschließlich der Ports 80 und 443, sofern die ausgewählten Ports nicht von einem anderen TCP/IP-Listener verwendet werden, der auf dem gleichen Host ausgeführt wird.

### **LocalAddress**

Die IP-Adresse, mit der alle Verbindungen zu dieser Route auf diesem Computer gebunden werden sollen. Die ausgewählte Adresse muss eine IP-Adresse sein, die einer der Netzchnittstellen auf dem Computer zugeordnet ist, auf dem MQIPT ausgeführt wird. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

### **MaxConnectionThreads**

Die maximale Anzahl der Verbindungsthreads und somit die maximale Anzahl gleichzeitiger Verbindungen, die von dieser Route verarbeitet werden können. Wenn dieser Grenzwert erreicht ist, gibt der Wert **MaxConnectionThreads** auch die Anzahl der Verbindungen an, die sich in der Warteschlange befinden, wenn alle Threads verwendet werden. Wenn diese Zahl erreicht wird, werden nachfolgende Verbindungsanforderungen abgelehnt. Der zulässige Mindestwert ist der größere Wert von 1 und der Wert von **MinConnectionThreads**. Wenn sich eine Änderung dieser Eigenschaft auf eine Route auswirkt, wird bei der Ausgabe eines Aktualisierungsbefehls der neue Wert verwendet. Alle Verbindungen verwenden sofort den neuen Wert. Die Route wird nicht gestoppt.

### **MinConnectionThreads**

Die Anzahl der Verbindungsthreads, die für die Verarbeitung eingehender Verbindungen auf einer Route zugeordnet sind, wenn die Route gestartet wird. Die Anzahl der zugeordneten Threads sinkt nicht unter diesen Wert ab, solange die Route aktiv ist. Der zulässige Mindestwert ist der kleinere

der beiden Werte `0` und **MaxConnectionThreads**. Änderungen an dieser Eigenschaft werden nur bei einem Neustart der Route wirksam.

### Name

Ein Name, mit dem die Route angegeben wird. Diese Eigenschaft ist optional. Der Wert wird in Konsolennachrichten und Traceinformationen angezeigt. Änderungen an dieser Eigenschaft werden nur bei einem Neustart der Route wirksam.

### OutgoingPort

Die Nummer des Startports, der von abgehenden Verbindungen verwendet wird. Der Bereich der Portnummern stimmt mit dem Wert **MaxConnectionThread** für diese Route überein. Bei Angabe des Standardwerts `0` wird eine vom System definierte Portnummer verwendet. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu dieser Route werden gestoppt. Wenn HTTP verwendet wird, sind für jede Kanalverbindung zwei abgehende Ports erforderlich. Weitere Informationen finden Sie im Abschnitt [Portnummernsteuerung](#).

### QMGrAccess

Setzen Sie **QMGrAccess** auf `Wahr`, um eingehende Kanalverbindungen des Warteschlangenmanagers (z. B. Senderkanäle) zuzulassen. Wenn Sie den Wert dieser Eigenschaft in `false` ändern, wird die Route gestoppt, wenn ein Befehl zur Aktualisierung ausgegeben wird. Alle Verbindungen zu dieser Route werden gestoppt.

### RouteRestart

Setzen Sie **RouteRestart** auf `Falsch`, damit die Route nicht erneut gestartet wird, wenn andere Routeneigenschaften geändert wurden und ein Aktualisierungsbefehl abgesetzt wurde. Der Standardwert für diese Eigenschaft ist `Wahr`.

### SecurityExit

Setzen Sie **SecurityExit** auf `Wahr`, um einen benutzerdefinierten Sicherheitsexit zu aktivieren. Der Standardwert für diese Eigenschaft ist `false`.

### SecurityExitName

Der Klassenname des benutzerdefinierten Sicherheitsexits. Diese Eigenschaft muss festgelegt werden, wenn **SecurityExit** auf `Wahr` gesetzt wurde. Wenn Sie diese Eigenschaft ändern (und **SecurityExit** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

### SecurityExitPath

Der vollständig qualifizierte Pfad mit dem benutzerdefinierten Sicherheitsexit. Wenn diese Eigenschaft nicht festgelegt wurde, wird standardmäßig das Unterverzeichnis des Exits verwendet. Diese Eigenschaft kann auch den Namen der Datei mit einem Java-Archiv (JAR-Datei) definieren, in dem sich der benutzerdefinierte Sicherheitsexit befindet. Wenn Sie diese Eigenschaft ändern (und **SecurityExit** auf `Wahr` gesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

### SecurityExitTimeout

Der Zeitlimitwert (in Sekunden), der von MQIPT verwendet wird, um festzulegen, wie lange auf eine Antwort gewartet wird, wenn eine Verbindungsanforderung validiert wird. Der Standardwert ist `30`. Wenn Sie diese Eigenschaft ändern (und **SecurityExit** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### SocksClient

Setzen Sie **SocksClient** auf `Wahr`, damit die Route als SOCKS-Client agiert und alle Verbindungen über den SOCKS-Proxy mit den Eigenschaften **SocksProxyHost** und **SocksProxyPort** definiert werden. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt. Diese Eigenschaft kann mit den folgenden Eigenschaften verwendet werden:

- **HTTP**
- **SocksServer**
- **SSLClient**

- **SSLProxyMode**

### **SocksProxyHost**

Der Hostname (oder die IPv4-Adresse in der Schreibweise mit Trennzeichen) des SOCKS-Proxys, den alle Verbindungen für diese Route verwenden. Wenn Sie diese Eigenschaft ändern (und **SocksClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt. Bei Verwendung der Eigenschaft **SocksProxyHost** muss die Eigenschaft **Destination** das Format in der Schreibweise mit Trennzeichen verwenden.

### **SocksProxyPort**

Die Portnummer, die in einem SOCKS-Proxy verwendet werden soll. Der Standardwert ist 1080. Wenn Sie diese Eigenschaft ändern (und **SocksClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SocksServer**

Setzen Sie **SocksServer** auf `Wahr`, damit die Route als SOCKS-Proxy agiert und SOCKS-Clientverbindungen akzeptiert. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt. Diese Eigenschaft kann mit den folgenden Eigenschaften verwendet werden:

- **SocksClient**
- **SSLProxyMode**
- **SSLServer**

### **SSLClient**

Setzen Sie **SSLClient** auf `Wahr`, damit die Route als SSL/TLS-Client agiert und abgehende SSL/TLS-Verbindungen hergestellt werden. Die Einstellung **SSLClient** auf `Wahr` impliziert, dass das Ziel entweder eine andere Instanz von MQIPT ist, die als SSL/TLS-Server fungiert, oder ein HTTP-Proxy/-Server.

Wenn Sie **SSLClient** auf `Wahr` setzen, müssen Sie einen SSL/TLS-Clientschlüsselring mit der Eigenschaft **SSLClientKeyRing** oder **SSLClientCAKeyRing** angeben oder konfigurieren Sie MQIPT für die Verwendung von Verschlüsselungshardware, indem Sie die Eigenschaft **SSLClientKeyRingUseCryptoHardware** oder **SSLClientCAKeyRingUseCryptoHardware** festlegen..

Wenn Sie **SSLClient** ändern, wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

Diese Eigenschaft kann nicht in Verbindung mit der folgenden Eigenschaft verwendet werden:

- **SSLProxyMode**

### **SSLClientCAKeyRing**

Der vollständig qualifizierte Dateiname der Schlüsselringdatei mit CA-Zertifikaten, die für die Authentifizierung von Zertifikaten vom SSL/TLS-Server verwendet werden. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientCAKeyRingPW**

Das Kennwort zum Öffnen der CA-Schlüsselringdatei des SSL/TLS-Clients, die mit der Eigenschaft **SSLClientCAKeyRing** angegeben wird, oder zum Herstellen einer Verbindung zum Schlüssel Speicher der Verschlüsselungshardware, wenn die Eigenschaft **SSLClientCAKeyRingUseCryptoHardware** auf `Wahr` gesetzt ist.

**V 9.1.5** Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptPW**

erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **V9.1.4** **SSLClientCAKeyRingUseCryptoHardware**

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher für CA-Zertifikate verwendet wird, die zur Authentifizierung von Serverzertifikaten vom SSL/TLS-Server verwendet werden, wenn MQIPT als SSL/TLS-Client agiert. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLClientCAKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Damit Sie diese Funktion verwenden können, muss der lokale Warteschlangenmanager, der über die MQIPT-Route verbunden ist, außerdem über die Berechtigung für IBM MQ Advanced, IBM MQ Appliance oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `Wahr` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** ist festgelegt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

#### **SSLClientCipherSuites**

Der Name der SSL/TLS-CipherSuite, die auf der SSL/TLS-Clientseite verwendet werden soll. Dabei kann es sich um eine oder mehrere der unterstützten CipherSuites handeln. Wenn Sie diese Eigenschaft leer lassen, verwendet der SSL/TLS-Client die unterstützten CipherSuites aus dem Schlüsselring. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

#### **SSLClientConnectTimeout**

Die Zeit in Sekunden, die ein SSL/TLS-Client wartet, bis eine SSL/TLS-Verbindung akzeptiert wird. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_C**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Ländernamen übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Ländernamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_CN**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem allgemeinen Namen übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_DC**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Domänenkomponente übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können

mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domänennamen dar. Der Domänenname `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Domänenkomponenten akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_DNQ**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Qualifikationsmerkmal für eine Domäne übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Qualifikationsmerkmale für Domänen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_L**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Position übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Positionen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_O**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Unternehmen übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate von allen Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_OU**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Organisationseinheit (Organizational Unit, OU) übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

#### **SSLClientDN\_PC**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Postleitzahl übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Postleitzahlen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientDN\_ST**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Status übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft



nicht angeben, werden Zertifikate in jedem Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientDN\_Street**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Straßennamen übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientDN\_T**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit diesem Titel übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientDN\_UID**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Server empfangen werden, welche mit dieser Benutzer-ID übereinstimmen. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden alle Benutzer-IDs akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientExit**

Mit dieser Eigenschaft können Sie die Verwendung eines Exits aktivieren oder inaktivieren, wenn die Route als SSL/TLS-Client fungiert. Auf diese Weise können Sie die Exitdetails in der Konfigurationsdatei definieren, ohne dass sie tatsächlich verwendet werden.

### **SSLClientKeyRing**

Der vollständig qualifizierte Dateiname der Schlüsselringdatei, die das Clientzertifikat enthält. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie **SSLClientKeyRing** ändern (und **SSLClient** auf `Wahr` gesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientKeyRingPW**

Das Kennwort zum Öffnen der mit der Eigenschaft **SSLClientKeyRing** angegebenen SSL/TLS-Clientschlüsselringdatei oder zum Herstellen einer Verbindung zum Keystore der Verschlüsselungshardware, wenn die Eigenschaft **SSLClientKeyRingUseCryptoHardware** auf `Wahr` gesetzt ist.

**V 9.1.5** Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptPW** erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Wenn Sie **SSLClientKeyRingPW** ändern (und **SSLClient** auf `Wahr` gesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

**V 9.1.4** **SSLClientKeyRingUseCryptoHardware**



Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher mit dem Clientzertifikat verwendet wird, wenn MQIPT als SSL/TLS-Client agiert. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLClientKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Damit Sie diese Funktion verwenden können, muss der lokale Warteschlangenmanager, der über die MQIPT-Route verbunden ist, außerdem über die Berechtigung für IBM MQ Advanced, IBM MQ Appliance oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `Wahr` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** ist festgelegt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

### SSLClientProtocols

Wird verwendet, um die Gruppe aktivierter Secure Socket-Protokolle zu beschränken, die verwendet werden, um abgehende Verbindungen zum Ziel für eine Route herzustellen, wenn **SSLClient** auf `Wahr` gesetzt ist.

Sie können mehrere Werte angeben, indem Sie sie mit Kommas trennen. V9.1.4 Wenn Sie diese Eigenschaft nicht angeben, ist das einzige standardmäßig aktivierte Protokoll TLS 1.2. Um andere Protokolle als TLS 1.2 zu aktivieren, müssen Sie in dieser Eigenschaft die Protokolle angeben, die aktiviert werden sollen, und Sie müssen Unterstützung für das Protokoll in der Java runtime environment hinzufügen und dabei die Vorgehensweise im Abschnitt [Veraltete Protokolle und CipherSuites](#) aktivieren befolgen. Sie können einen oder mehrere der folgenden Werte angeben.

<i>Tabelle 51. Zulässige Werte für SSL/TLS-Protokolle</i>	
<b>Wert</b>	<b>Protokoll</b>
SSLv3	SSL 3.0
TLSv1	TLS 1.0
TLSv1.1	TLS 1.1
TLSv1.2	TLS 1.2

Verwenden Sie den Eintrag, der in der Spalte **Value** (Wert) der Routeneigenschaft angegeben ist. Der zugehörige Eintrag in der Spalte **Protocol** (Protokoll) dient nur zur Information.

### SSLClientSiteDN\_C

Verwenden Sie diese Eigenschaft zur Angabe eines Ländernamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Ländernamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### SSLClientSiteDN\_CN

Verwenden Sie diese Eigenschaft zur Angabe eines allgemeinen Namens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### SSLClientSiteDN\_DC

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für die Domänenkomponente, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstim-

mungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domänennamen dar. Der Domänenname `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Domänenkomponente akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientSiteDN\_DNQ**

Verwenden Sie diese Eigenschaft zur Angabe einer Domäne für ein Qualifikationsmerkmal, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Qualifikationsmerkmal akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientSiteDN\_L**

Verwenden Sie diese Eigenschaft zur Angabe eines Positionsnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Position akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientSiteDN\_O**

Verwenden Sie diese Eigenschaft zur Angabe eines Unternehmensnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientSiteDN\_OU**

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für eine Organisationseinheit (OU), mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

#### **SSLClientSiteDN\_PC**

Verwenden Sie diese Eigenschaft zur Angabe einer Postleitzahl, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Postleitzahl akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLClientSiteDN\_ST**

Verwenden Sie diese Eigenschaft zur Angabe eines Statusnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für den Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClientauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientSiteDN\_Street**

Verwenden Sie diese Eigenschaft zur Angabe eines Straßennamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientSiteDN\_T**

Verwenden Sie diese Eigenschaft zur Angabe eines Titels, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientSiteDN\_UID**

Verwenden Sie diese Eigenschaft zur Angabe einer Benutzer-ID, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Benutzer-ID akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLClientSiteLabel**

Verwenden Sie diese Eigenschaft zur Angabe einer Bezeichnung, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Server gesendet werden soll. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Bezeichnung akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLClient** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLExitData**

Verwenden Sie diese Eigenschaft, um eine benutzerdefinierte Zeichenfolge bereitzustellen, die an den Exit übergeben werden soll.

### **SSLExitName**

Mit dieser Eigenschaft können Sie den Klassennamen für den Exit definieren, der aufgerufen wird, wenn eine Route als SSL/TLS-Client oder SSL/TLS-Server fungiert. Der Name muss einen beliebigen Paketnamen enthalten, z. B. `com.ibm.mq.ipc.exit.TestExit`.

### **SSLExitPath**

Mit dieser Eigenschaft können Sie die Position des Exits definieren, der zum Laden einer Kopie des Exits verwendet wird. Der Name muss ein vollständig qualifizierter Name sein, der für die Suche nach der Klassendatei oder dem Namen einer `.jar`-Datei mit der Klassendatei verwendet werden kann, wie beispielsweise `C:\mqipt\exits` oder `C:\mqipt\exits\exits.jar`.

### **SSLExitTimeout**

Definieren Sie mit dieser Eigenschaft, wie lange MQIPT auf den Abschluss des Exits wartet, bevor die Verbindungsanforderung beendet wird. Der Wert `0` gibt an, dass MQIPT unendlich wartet.

### **SSLPlainConnections**

Mit dieser Eigenschaft können Sie angeben, ob SSL/TLS für Verbindungen zum MQIPT-Listener-Port einer Route verbindlich ist, die für das Akzeptieren eingehender SSL/TLS-Verbindungen konfiguriert ist. Diese Eigenschaft gilt für Routen, für die die Eigenschaft **SSLServer** oder **SSLProxyMode** auf `"true"` gesetzt ist. Wenn diese Eigenschaft aktiviert ist, können unverschlüsselte Verbindungen zum Listener-Port der Route hergestellt werden, wodurch MQIPT alle IBM MQ-Verbindungen zum Listener-Port des Warteschlangenmanagers weiterleiten kann, unabhängig davon, ob die Verbindung verschlüsselt ist. Wenn Sie diesen Parameter nicht festlegen oder ihn auf `false` setzen, sind nur eingehende SSL/TLS-Verbindungen zulässig. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

### SSLProxyMode

Setzen Sie diese Eigenschaft auf `true`, damit die Route nur SSL/TLS-Clientverbindungsanforderungen akzeptiert und die Anforderung direkt an das Ziel übertragen wird. Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu dieser Route werden gestoppt. Diese Eigenschaft kann nicht in Verbindung mit den folgenden Eigenschaften verwendet werden:

- **SocksClient**
- **SocksServer**
- **SSLClient**
- **SSLServer**

### SSLServer

Setzen Sie diese Eigenschaft auf `true`, damit die Route als SSL/TLS-Server fungiert und eingehende SSL/TLS-Verbindungen akzeptiert. Wenn Sie **SSLServer** auf `Wahr` setzen, bedeutet dies, dass der Aufrufende ein anderer MQIPT ist, der als SSL/TLS-Client fungiert, oder ein IBM MQ-Client oder -Warteschlangenmanager mit aktiviertem SSL/TLS.

Wenn Sie **SSLServer** auf `Wahr` setzen, müssen Sie einen SSL/TLS-Serverschlüsselring angeben mit Hilfe der **SSLServerKeyRing**-Eigenschaft oder konfigurieren Sie MQIPT für die Verwendung von Verschlüsselungshardware, indem Sie die Eigenschaft **SSLServerKeyRingUseCryptoHardware** festlegen..

Wenn Sie diese Eigenschaft ändern, wird die Route gestoppt und bei der Ausgabe eines Aktualisierungsbefehls erneut gestartet. Alle Verbindungen zu der Route werden gestoppt.

Diese Eigenschaft kann nicht in Verbindung mit den folgenden Eigenschaften verwendet werden:

- **SocksServer**
- **SSLProxyMode**

### SSLServerCAKeyRing

Der vollständig qualifizierte Dateiname der Schlüsselringdatei mit CA-Zertifikaten, die für die Authentifizierung von Zertifikaten vom SSL/TLS-Client verwendet werden. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

### SSLServerCAKeyRingPW

Das Kennwort zum Öffnen der CA-Schlüsselringdatei des SSL/TLS-Servers, die mit der Eigenschaft **SSLServerCAKeyRing** angegeben ist, oder zum Herstellen einer Verbindung zum Schlüsselspeicher der Verschlüsselungshardware, wenn die Eigenschaft **SSLServerCAKeyRingUseCryptoHardware** auf `Wahr` gesetzt ist

**V 9.1.5** Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptPW** erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **V 9.1.4** SSLServerCAKeyRingUseCryptoHardware

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher für die CA-Zertifikate verwendet wird, die zur Authentifizierung von Zertifikaten vom SSL/TLS-Client

ent verwendet werden. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLServerCAKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Damit Sie diese Funktion verwenden können, muss der lokale Warteschlangenmanager, der über die MQIPT-Route verbunden ist, außerdem über die Berechtigung für IBM MQ Advanced, IBM MQ Appliance oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `Wahr` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** ist festgelegt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

### **SSLServerAskClientAuth**

Verwenden Sie diese Eigenschaft für die Anforderung der SSL/TLS-Clientauthentifizierung durch den SSL/TLS-Server. Der SSL/TLS-Client muss über ein eigenes Zertifikat verfügen, das an den SSL/TLS-Server gesendet wird. Das Zertifikat wird aus der Schlüsselringdatei abgerufen. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

### **SSLServerCipherSuites**

Der Name der SSL/TLS-CipherSuite, die auf der Seite des SSL/TLS-Servers verwendet werden soll. Dabei kann es sich um eine oder mehrere der unterstützten CipherSuites handeln. Wenn Sie diese Eigenschaft leer lassen, verwendet der SSL/TLS-Server die unterstützten CipherSuites aus dem Schlüsselring. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

### **SSLServerDN\_C**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Ländernamen empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Unternehmensnamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_CN**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem allgemeinen Namen empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_DC**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dem Namen dieser Domänenkomponente empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domännennamen dar. Der Domänenname `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Domänenkomponente akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_DNQ**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Qualifikationsmerkmal für die Domäne empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Qualifikationsmerkmal akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_L**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dieser Position empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Position akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_O**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Unternehmen empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_OU**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client dieser Organisationseinheit (OU) empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

### **SSLServerDN\_PC**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dieser Postleitzahl empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Postleitzahl akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_ST**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Status empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### **SSLServerDN\_Street**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Straßennamen empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden

Zertifikate mit einem beliebigen Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerDN\_T**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit diesem Titel empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerDN\_UID**

Verwenden Sie diese Eigenschaft zum Akzeptieren von Zertifikaten, die vom SSL/TLS-Client mit dieser Benutzer-ID empfangen werden. Der Name kann mit einem Stern (\*) als Präfix oder Suffix versehen sein, um den Geltungsbereich zu erweitern. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Benutzer-ID akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerExit**

Mit dieser Eigenschaft können Sie die Verwendung eines Exits aktivieren oder inaktivieren, wenn die Route als SSL/TLS-Server fungiert. Auf diese Weise können Sie die Exitdetails in der Konfigurationsdatei definieren, ohne dass sie tatsächlich verwendet werden.

#### **SSLServerKeyRing**

Der vollständig qualifizierte Dateiname der Schlüsselringdatei, die das Serverzertifikat enthält. Auf Windows-Plattformen müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerKeyRingPW**

Das Kennwort zum Öffnen der mit der Eigenschaft **SSLServerKeyRing** angegebenen SSL/TLS-Serververschlüsselungsdatei oder zum Herstellen einer Verbindung zum Schlüsselspeicher der Verschlüsselungshardware, wenn die Eigenschaft **SSLServerKeyRingUseCryptoHardware** auf `Wahr` gesetzt ist.

**V 9.1.5** Bei dem Wert kann es sich um ein Kennwort handeln, das mit dem Befehl **mqiptPW** verschlüsselt wurde, oder um den vollständig qualifizierten Dateinamen der Datei mit einem verschlüsselten Kennwort. Wenn Sie auf Windows-Plattformen einen Dateinamen angeben, müssen Sie einen doppelten Backslash (\\) als Dateitrennzeichen verwenden. Es wird empfohlen, alle Schlüsselringkennwörter, die aktuell in einer Datei gespeichert, für die Verwendung der neuesten und sichersten Zugriffsschutzmethode zu migrieren, indem die Kennwörter mit dem Dienstprogramm **mqiptPW** erneut verschlüsselt werden. Weitere Informationen zur Verschlüsselung von Kennwörtern in der MQIPT-Konfiguration finden Sie unter [Gespeicherte Kennwörter verschlüsseln](#).

Sie müssen **SSLServerKeyRingPW** angeben, wenn Sie **SSLServer** auf `Wahr` setzen.

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **V 9.1.4 SSLServerKeyRingUseCryptoHardware**

Gibt an, ob Verschlüsselungshardware, die die PKCS#11-Schnittstelle unterstützt, als Schlüsselspeicher für das Serverzertifikat verwendet wird, wenn MQIPT als SSL/TLS-Server agiert. Wenn diese Eigenschaft auf `Wahr` gesetzt ist, kann **SSLServerKeyRing** nicht auf derselben Route festgelegt werden.

Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

Die Verwendung von Verschlüsselungshardware mit MQIPT ist eine Funktionalität von IBM MQ Advanced. Damit Sie diese Funktion verwenden können, muss der lokale Warteschlangenmanager, der über die MQIPT-Route verbunden ist, außerdem über die Berechtigung für IBM MQ Advanced, IBM MQ Appliance oder IBM MQ Advanced for z/OS VUE verfügen. Die Route wird nicht gestartet, wenn diese Eigenschaft auf `Wahr` gesetzt ist, es sei denn, die globale Eigenschaft **EnableAdvancedCapabilities** ist festgelegt, um zu bestätigen, dass IBM MQ Advanced-Funktionen verwendet werden können.

### SSLServerProtocols

Wird verwendet, um die Gruppe aktivierter Secure Socket-Protokolle zu beschränken, die verwendet werden, um eingehende Verbindungen zum Route-Listener-Port für eine Route zu akzeptieren, wenn **SSLServer** auf `Wahr` gesetzt ist.

Sie können mehrere Werte angeben, indem Sie sie mit Kommas trennen. V9.1.4 Wenn Sie diese Eigenschaft nicht angeben, ist das einzige standardmäßig aktivierte Protokoll TLS 1.2. Um andere Protokolle als TLS 1.2 zu aktivieren, müssen Sie in dieser Eigenschaft die Protokolle angeben, die aktiviert werden sollen, und Sie müssen Unterstützung für das Protokoll in der Java runtime environment hinzufügen und dabei die Vorgehensweise im Abschnitt [Veraltete Protokolle und CipherSuites](#) aktivieren befolgen. Sie können einen oder mehrere der folgenden Werte angeben.

<i>Tabelle 52. Zulässige Werte für SSL/TLS-Protokolle</i>	
<b>Wert</b>	<b>Protokoll</b>
SSLv3	SSL 3.0
TLSv1	TLS 1.0
TLSv1.1	TLS 1.1
TLSv1.2	TLS 1.2

Verwenden Sie den Eintrag, der in der Spalte **Value** (Wert) der Routeneigenschaft angegeben ist. Der zugehörige Eintrag in der Spalte **Protocol** (Protokoll) dient nur zur Information.

### SSLServerSiteDN\_C

Verwenden Sie diese Eigenschaft zur Angabe eines Ländernamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Ländernamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### SSLServerSiteDN\_CN

Verwenden Sie diese Eigenschaft zur Angabe eines allgemeinen Namens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen allgemeinen Namen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf `Wahr` gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

### SSLServerSiteDN\_DC

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für die Domänenkomponente, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie können mehrere Domänenkomponenten angeben, indem Sie sie mit Kommas trennen. Jede Domänenkomponente stellt ein Element in einem Domännennamen dar. Der Domänenname `example.ibm.com` wird beispielsweise als `example,ibm,com` dargestellt, in dem die verschiedenen Werte durch Kommas getrennt werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen



für die Domänenkomponente akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_DNQ**

Verwenden Sie diese Eigenschaft zur Angabe einer Domäne für ein Qualifikationsmerkmal, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Qualifikationsmerkmal akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_L**

Verwenden Sie diese Eigenschaft zur Angabe eines Positionsnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Position akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_O**

Verwenden Sie diese Eigenschaft zur Angabe eines Unternehmensnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für das Unternehmen akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_OU**

Verwenden Sie diese Eigenschaft zur Angabe eines Namens für eine Organisationseinheit (OU), mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Sie können mehrere Organisationseinheiten angeben, indem Sie sie mit Kommas trennen. (Gleichen Sie ein Literalkomma ab, indem Sie ein Backslash (\) als Präfix voranstellen.) Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für die Organisationseinheit akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu dieser Route werden gestoppt.

#### **SSLServerSiteDN\_PC**

Verwenden Sie diese Eigenschaft zur Angabe einer Postleitzahl, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Postleitzahl akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_ST**

Verwenden Sie diese Eigenschaft zur Angabe eines Statusnamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Namen für den Status akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServerauf** Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_Street**

Verwenden Sie diese Eigenschaft zur Angabe eines Straßennamens, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Straßennamen akzeptiert. Wenn Sie diese Eigenschaft ändern (und

**SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_T**

Verwenden Sie diese Eigenschaft zur Angabe eines Titels, mit dem ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einem beliebigen Titel akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteDN\_UID**

Verwenden Sie diese Eigenschaft zur Angabe einer Benutzer-ID, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Bei Übereinstimmungen von Zertifikaten muss die Groß-/Kleinschreibung nicht beachtet werden. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Benutzer-ID akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **SSLServerSiteLabel**

Verwenden Sie diese Eigenschaft zur Angabe einer Bezeichnung, mit der ein Zertifikat ausgewählt wird, das an den SSL/TLS-Client gesendet werden soll. Wenn Sie diese Eigenschaft nicht angeben, werden Zertifikate mit einer beliebigen Bezeichnung akzeptiert. Wenn Sie diese Eigenschaft ändern (und **SSLServer** auf Wahrgesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird. Alle Verbindungen zu der Route werden gestoppt.

#### **V 9.1.5**

#### **StoredCredentialsFormat**

Verwenden Sie diese Eigenschaft, um anzugeben, ob die Werte von Kennworteigenschaften das verschlüsselte Kennwortformat verwenden, das von MQIPT in IBM MQ 9.1.5 unterstützt wird. MQIPT kann fast immer erkennen, ob Kennwörter im Format für verschlüsselte Kennwörter angegeben sind. Diese Eigenschaft muss nur in dem unwahrscheinlichen Fall festgelegt werden, in dem MQIPT nicht automatisch zwischen einem verschlüsselten Kennwort und einem Klartextkennwort oder einem Dateinamen unterscheiden kann.

Dieser kann einen der folgenden Werte annehmen:

#### **encrypted**

Kennworteigenschaften enthalten ein verschlüsseltes Kennwort in dem Format, das von MQIPT in IBM MQ 9.1.5 unterstützt wird.

#### **compat**

Kennworteigenschaften enthalten entweder ein Klartextkennwort oder bei Schlüsselringkennwörtern den Namen der Datei, die ein verschlüsseltes Kennwort enthält.

#### **TCPKeepAlive**

Setzen Sie diese Eigenschaft auf `true`, um das regelmäßige Senden von TCP/IP-Keepalive-Paketen zu aktivieren, mit denen verhindert wird, dass Verbindungen in dieser Route inaktiv werden. Dies verringert die Wahrscheinlichkeit, dass MQIPT-Verbindungen durch eine Firewall oder einen Router getrennt werden. Das Senden von TCP/IP-Keepalive-Paketen wird durch Optimierungsparameter des Betriebssystems gesteuert. Weitere Informationen zum Einstellen von Keepalive-Paketen finden Sie in der Dokumentation zu Ihrem Betriebssystem. Wenn Sie diesen Parameter nicht festlegen oder ihn auf `false` setzen, werden keine Keepalive-Pakete gesendet.

#### **Trace**

Die für diese Route erforderliche Tracestufe. Durch das Aktivieren der Tracefunktion für eine Route wird die Tracefunktion gleichzeitig nicht für andere Routen aktiviert. Wenn Sie mehrere Routen verfolgen müssen, müssen Sie die Eigenschaft **Trace** zum Abschnitt `[route]` jeder Route hinzufügen, für die ein Trace erstellt werden soll.

Diese Eigenschaft kann einen der folgenden Werte haben:

#### **0**

Trace ist nicht aktiviert

### Jede beliebige positive Ganzzahl.

Trace ist aktiviert

Der Standardwert ist 0.

Wenn der Abschnitt [route] keine Eigenschaft **Trace** enthält, wird die Eigenschaft **Trace** aus dem Abschnitt [global] verwendet. Weitere Informationen zur Tracefunktion für Threads, die keiner Route zugeordnet sind, finden Sie unter **Trace** im Abschnitt [global]. Wenn sich eine Änderung dieser Eigenschaft auf eine Route auswirkt, wird bei der Ausgabe eines Aktualisierungsbefehls der neue Wert verwendet. Alle Verbindungen verwenden sofort den neuen Wert. Die Route wird nicht gestoppt.

### UriName

Mit dieser Eigenschaft kann der Name eines Uniform Resource Identifier (URI) der Ressource bei der Verwendung eines HTTP-Proxys geändert werden, obwohl der Standardwert für die meisten Konfigurationen ausreicht:

```
HTTP://destination:destination_port/mqipt
```

Wenn Sie diese Eigenschaft ändern (und **HTTPOff** auf **Wahr** gesetzt ist), dann wird die Route gestoppt und erneut gestartet, wenn ein Aktualisierungsbefehl abgesetzt wird.

## z/OS V 9.1.0 mqzOSConnectService -Element

Der MQ Service Provider wird als Liberty-Standardfeature bereitgestellt und daher mit server.xml konfiguriert. Jeder uni- oder bidirektionale Service wird in einem Element mqzOSConnectService definiert.

**Wichtig:** Ein Element mqzOSConnectService kann erst verwendet werden, wenn von einem Element zOSConnectService darauf verwiesen wird.

Im Folgenden ist ein Beispiel für ein Element mqzOSConnectService mit Angabe einiger Attribute dargestellt.

```
<mqzOSConnectService id="twoWay "
  connectionFactory="jms/cf1"
  destination="jms/requestQueue"
  replyDestination="jms/replyQueue"
  expiry="-1"
  waitInterval="10000"
  replySelection="msgIDToCorrelID"
  selector=""
  persistence="false"/>
```



**Achtung:** Je nachdem, wie der MQ Service Provider installiert wurde, ist dem Element mqzOS-ConnectService möglicherweise eine Zeichenfolge, gefolgt von einem Unterstrich, als Präfix vorangestellt. Beispiel: `usr_mqzOSConnectService`).

Dies wird im Abschnitt [MQ Service Provider in IBM z/OS Connect EE installieren](#) für z/OS Connect EE beschrieben.

Bei dem folgenden Beispiel wurde der MQ Service Provider im WLP-Kernel installiert.

Attributname	Typ	Standardwert	Beschreibung
ID	Zeichenfolge		„ID“ auf Seite 284
connectionFactory	Ein JNDI-Name (Zeichenfolge).		„connectionFactory“ auf Seite 284

Tabelle 53. Attribute des Elements `mqzOSConnectService` (Forts.)

Attributname	Typ	Standardwert	Beschreibung
destination	Ein JNDI-Name (Zeichenfolge).		„destination“ auf Seite 284
replyDestination	Ein JNDI-Name (Zeichenfolge).		„replyDestination“ auf Seite 285
expiry	integer	-1	„expiry“ auf Seite 285
waitInterval	integer		„waitInterval“ auf Seite 285
replySelection	Zeichenfolge	msgIDToCorrelID	„replySelection“ auf Seite 285
Selektor	Zeichenfolge		„Selektor“ auf Seite 286
persistence	boolean	false	„persistence“ auf Seite 286
mqmdFormat	Zeichenfolge		„mqmdFormat“ auf Seite 286
userName	Zeichenfolge		„userName“ auf Seite 287
Kennwort	Zeichenfolge		„Kennwort“ auf Seite 287
useCallerPrincipal	boolean	false	„useCallerPrincipal“ auf Seite 287
receiveTextCCSID	integer	37	„receiveTextCCSID“ auf Seite 287

## ID

**id** ist ein erforderliches Attribut, das für alle Elemente in `server.xml` eindeutig sein muss. **id** wird vom Element `zosConnectService` verwendet, um auf eine Ziel-Service-Provider-Instanz zu verweisen.

## connectionFactory

**connectionFactory** gibt den JNDI-Namen einer IBM MQ-Messaging-Provider-Verbindungsfactory an. Der MQ Service Provider verwendet die Verbindungsfactory, um eine Verbindung zu IBM MQ herzustellen.

**connectionFactory** ist ein erforderliches Attribut. Weitere Informationen zu Verbindungsfactorys finden Sie unter [JMS-Verbindungsfactory](#).

## destination

**destination** gibt den JNDI-Namen eines IBM MQ-Messaging-Provider-Ziels an.

**destination** ist ein erforderliches Attribut.

Weitere Konfigurationsinformationen:

- Konfiguration einer Warteschlange in WLP: siehe Abschnitt [JMS Queue](#) (JMS-Warteschlange).
- Konfiguration eines Topics in WLP: siehe Abschnitt [JMS Topic](#).

Bei einem unidirektionalen Service wird **destination** als Ziel für HTTP POST-, HTTP GET- und HTTP DELETE-Anforderungen verwendet.

Beachten Sie, dass Warteschlangenziele für alle drei Anforderungstypen unterstützt werden, während Topic-Ziele nur bei HTTP POST-Anforderungen unterstützt werden.

Bei einem bidirektionalen Service muss **destination** ein Warteschlangenziel sein, das die vom Back-End-Service verwendete Anforderungswarteschlange darstellt.

Bidirektionale Services unterstützen nur HTTP POST-Anforderungen.

## replyDestination

**replyDestination** gibt den JNDI-Namen einer IBM MQ-Messaging-Provider-Warteschlange an.

**replyDestination** ist ein optionales Attribut.

Weitere Informationen zum Konfigurieren einer Warteschlange in WLP finden Sie im Abschnitt [JMS Queue](#) (JMS-Warteschlange).

Wenn **replyDestination** nicht angegeben ist, handelt es sich um einen unidirektionalen Service. Ist **replyDestination** dagegen angegeben, ist der Service ein bidirektionaler Service.

Diese Warteschlange ist das Antwortziel, an das der Back-End-Service Antwortnachrichten sendet.

## expiry

**expiry** gibt an, wie lange Nachrichten, die vom MQ Service Provider gesendet wurden, gültig sind. Die Angabe erfolgt in tausendstel Sekunden ab dem Zeitpunkt des Sendens. Die Nachricht kann gelöscht werden, wenn sie nicht aus der Zielwarteschlange entfernt wird, bevor dieser Zeitraum verstrichen ist.

**expiry** ist ein optionales Attribut und entspricht der Einstellung des MQMD-Felds [Expiry](#).

Bei Angabe negativer Werte laufen die Nachrichten niemals ab. Der Standardwert für **expiry** lautet -1.

REST-Clients können **expiry** durch Angabe eines HTTP-Headers `ibm-mq-md-expiry` mit gültiger 64-Bit-Ganzzahl außer Kraft setzen.

## waitInterval

Bei HTTP DELETE-Anforderungen an unidirektionale Services gibt **waitInterval** die Anzahl der Millisekunden an, die der Service auf eine übereinstimmende Nachricht in der Warteschlange wartet, die über das Attribut [destination](#) angegeben wurde.

Bei HTTP POST-Anforderungen an bidirektionale Services gibt **waitInterval** die Anzahl der Millisekunden an, die der Service auf eine übereinstimmende Nachricht in der Warteschlange wartet, die über das Attribut [replydestination](#) angegeben wurde.

**waitInterval** ist ein optionales Attribut für unidirektionale Services, ein erforderliches Attribut für bidirektionale Services und entspricht der Einstellung des MQMD-Felds [WaitInterval](#).

**waitInterval** wird bei HTTP GET-Anforderungen nicht unterstützt.

Für **waitInterval** gilt Folgendes:

- Bei einem Wert von null wartet der Service nicht.
  - Bei bidirektionalen Services wird der Wert null für **waitInterval** nicht unterstützt.
- Bei einem negativen Wert wartet der Service unbegrenzt, bis eine Nachricht verfügbar ist.

REST-Clients können diesen Wert durch Angabe eines HTTP-Headers `ibm-mq-gmo-waitInterval` mit gültiger 64-Bit-Ganzzahl außer Kraft setzen.

**Anmerkung:** Wenn ein großer oder negativer Wert für **waitInterval** angegeben wird, wird es voraussichtlich zu Zeitlimitüberschreitungen bei den Transaktionen und asynchronen Serviceanforderungen kommen. Wenn eines oder beide diese Ereignisse eintreten, erhöhen Sie das Zeitlimit und/oder verringern Sie das Warteintervall.

## replySelection

**replySelection** beschreibt den Mechanismus zum Abgleich von Antwortnachrichten mit Anforderungsnachrichten.

replySelection ist optional und wird nur bei bidirektionalen Services verwendet. Wenn replySelection bei einem unidirektionalen Service verwendet wird, wird die Angabe ignoriert.

Folgende Werte sind möglich:

#### **msgIDToCorrelID**

Es wird angenommen, dass bei der Generierung von Antwortnachrichten für die Korrelations-ID der Wert der Nachrichten-ID der Anforderungsnachricht festgelegt wird. Der Service erstellt anhand dieser Informationen einen geeigneten Nachrichtenselektor. Dies ist der Standardwert.

#### **none**

Es wird kein Mechanismus zum Korrelieren von Antwortnachrichten mit Anforderungsnachrichten verwendet. Der Service ruft die erste verfügbare Nachricht in der Antwortwarteschlange ab.

#### **correlIDToCorrelID**

Es wird angenommen, dass bei der Generierung von Antwortnachrichten für die Korrelations-ID der Wert der Korrelations-ID der Anforderungsnachricht festgelegt wird. Der Service erstellt anhand dieser Informationen einen geeigneten Nachrichtenselektor. Wenn für die Anforderungsnachricht keine Korrelations-ID angegeben ist (siehe Abschnitt „[ibm-mq-md-correlID](#)“ auf Seite 288), generiert der Service eine zufällige Korrelations-ID für die Anforderungsnachricht.

## **Selektor**

selector muss ein gültiger JMS-Nachrichtensektor gemäß der Beschreibung in der JMS-Spezifikation sein.

selector wird nur bei unidirektionalen Services verwendet und ist optional. Wenn selector bei bidirektionalen Services verwendet wird, wird die Angabe ignoriert. Weitere Informationen zu Selektoren finden Sie im Abschnitt [Nachrichtenselektoren in JMS](#).

selector wird bei HTTP GET- und HTTP DELETE-Anforderungen zur Auswahl der zurückgegebenen Nachricht verwendet. Wenn die Header „[ibm-mq-md-msgID](#)“ auf Seite 288 oder „[ibm-mq-md-correlID](#)“ auf Seite 288 angegeben sind, wird selector ignoriert.

Manche für 'selector' verwendete Zeichen müssen codiert werden, um in die Datei server.xml integriert werden zu können. Hierfür können Sie folgende Standardmechanismen verwenden:

```
" becomes &quot;  
' becomes &apos;  
< becomes &lt;> becomes &gt;
```

## **persistence**

persistence gibt die Persistenz von Nachrichten an, die von einem Service gesendet wurden.

persistence ist optional und entspricht der Einstellung des MQMD-Felds [Persistence](#).

Folgende Werte sind möglich:

#### **false**

Die Nachrichten sind nicht persistent. Dies ist der Standardwert.

#### **true**

Die Nachrichten sind persistent.

persistence kann durch Verwendung eines HTTP-Headers `ibm-mq-md-persistence`, der dieselben Werte annimmt, außer Kraft gesetzt werden.

## **mqmdFormat**

Dieses Attribut wird verwendet, um den Wert des MQMD-Formatfeldes in Nachrichten festzulegen, die vom MQ Service Provider gesendet werden. Es wird allerdings nur verwendet, wenn der MQ Service Provider für die Verwendung von z/OS Connect-Datenkonvertierungen konfiguriert wurde. Andernfalls wird das Attribut ignoriert.

Wenn Sie dieses Attribut nicht angeben und Datenkonvertierungen verwendet werden, werden Nachrichten mit dem MQMD-Formatfeld gesendet, das auf "MQSTR " gesetzt ist. Der Wert für dieses Attribut darf maximal acht Zeichen umfassen.

## **userName**

Der Benutzername, den der MQ Service Provider IBM MQ zu Authentifizierungs- und Autorisierungszwecken angibt.

Wenn Sie dieses Attribut nicht angeben, wird das Attribut **userName** in der Verbindungsfactory verwendet, auf die über das Attribut **connectionFactory** verwiesen wird.

Wenn das Attribut **userName** sowohl in der referenzierten Verbindungsfactory als auch auf dem MQ Service Provider angegeben ist, wird der Wert des MQ Service Providers verwendet.

Wenn Sie dieses Attribut angeben, müssen Sie auch das Attribut **password** angeben.

## **Kennwort**

Das Kennwort, das der MQ Service Provider IBM MQ zu Authentifizierungs- und Autorisierungszwecken angibt.

Sie können das Kennwort als unverschlüsselter Text angeben, dies ist jedoch nicht zu empfehlen. Sie sollten das Kennwort mit dem Tool **securityUtility** codieren, das mit z/OS Connect bereitgestellt wird, und hierfür die Codierungsoption verwenden. Weitere Informationen finden Sie unter [Liberty: Befehl securityUtility](#).

Wenn Sie dieses Attribut nicht angeben, wird das Attribut 'password' in der Verbindungsfactory verwendet, auf die über das Attribut **connectionFactory** verwiesen wird.

Wenn das Attribut 'password' sowohl in der referenzierten Verbindungsfactory als auch auf dem MQ Service Provider angegeben wird, wird der Wert des MQ Service Providers verwendet.

Wenn Sie dieses Attribut angeben, müssen Sie auch das Attribut **userName** angeben.

## **useCallerPrincipal**

Wenn eine Anforderung an z/OS Connect gestellt wird, authentifiziert sich das aufrufende Programm bei z/OS Connect. Der Name des authentifizierten Principals kann zu Authentifizierungs- und Autorisierungszwecken an IBM MQ übergeben werden.

Legen Sie zu diesem Zweck für **useCallerPrincipal** den Wert 'true' fest.

Beim Herstellen einer Verbindung zu IBM MQ wird der Name des Principals, aber kein Kennwort verwendet. Alle Werte, die für die Attribute **password** und **userName** angegeben werden, werden ignoriert.

## **receiveTextCCSID**

Die CCSID, die verwendet wird, wenn eine Datenkonvertierung erhalten und `javax.jms.TextMessage` verarbeitet wird (d. h. HTTP GET oder HTTP DELETE bei einem unidirektionalen Service oder beim Abruf einer Antwortnachricht für einen bidirektionalen Service).

Der Text in der Nachricht wird in die über **receiveTextCCSID** angegebene CCSID umgewandelt.

## **HTTP-Header, die mit dem MQ Service Provider verwendet werden können**

Nur bei Ausgabe einer HTTP POST-Anforderung erwartet der MQ Service Provider bestimmte HTTP-Header.

In diesem Fall muss der Content-Type-Header auf "application/json" gesetzt sein. Wenn Sie als Teil dieses Headers einen Zeichensatz angeben, muss sein Wert utf-8 lauten.

Beispiel: Content-Type=application/json;charset=utf-8.

Andere HTTP-Header können in der HTTP-Anforderung angegeben werden, um das Verhalten des MQ Service Providers zu ändern. Diese werden in den folgenden Abschnitten erläutert. Alle anderen HTTP-Header werden ignoriert.

### **ibm-mq-md-msgID**

Dieser Header kann angegeben werden, wenn HTTP GET- oder HTTP DELETE-Anforderungen für unidirektionale Services ausgegeben werden.

Der Wert dieses Headers wird verwendet, um einen Nachrichtenselektor zu generieren und somit eine Nachricht mit der angegebenen Nachrichten-ID auszuwählen. Wenn auch ein „[ibm-mq-md-correlID](#)“ auf [Seite 288-Header](#) angegeben ist, wird ein Nachrichtenselektor generiert, der beide IDs abgleicht.

### **ibm-mq-md-correlID**

Dieser Header kann angegeben werden, wenn eine HTTP POST-Anforderung ausgegeben wird. In diesem Fall wird er zur Definition des MQMD-Felds [CorrelID](#) der Nachricht verwendet, die gesendet wird.

Dieser Header kann auch angegeben werden, wenn HTTP GET- oder HTTP DELETE-Anforderungen für unidirektionale Services ausgegeben werden. Der Wert dieses Headers wird verwendet, um einen Nachrichtenselektor zu generieren und somit eine Nachricht mit der angegebenen Korrelations-ID auszuwählen. Wenn auch ein „[ibm-mq-md-msgID](#)“ auf [Seite 288-Header](#) angegeben ist, wird ein Nachrichtenselektor generiert, der beide abgleicht.

### **ibm-mq-pmo-retain**

Sie können diesen Header mit dem Wert TRUE angeben, wenn Sie eine HTTP POST-Anforderung gestützt durch ein Topic an einen unidirektionalen Service ausgeben. Es wird dann eine ständige Veröffentlichung generiert. Weitere Informationen finden Sie im Abschnitt [Ständige Veröffentlichungen](#).

### **ibm-mq-usr**

Sie können diesen Header verwenden, um Nachrichteneigenschaften in den IBM MQ-Nachrichten anzugeben, die infolge von HTTP POST-Anforderungen an unidirektionale und bidirektionale Services gesendet werden.



## Bemerkungen

---

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieser Dokumentation ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe  
IBM Europe, Middle East and Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Défense  
U.S.A.

Bei Lizenzanforderungen zu Double-Byte-Information (DBCS) wenden Sie sich bitte an die IBM Abteilung für geistiges Eigentum in Ihrem Land oder senden Sie Anfragen schriftlich an folgende Adresse:

Lizenzierung von geistigem Eigentum

IBM Japan, Ltd.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in dieser Veröffentlichung werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Europe, Middle East and Africa  
Software Interoperability Coordinator, Department 49XA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesen Informationen beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Die in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Um diese so realistisch wie möglich zu gestalten, enthalten sie auch Namen von Personen, Firmen, Marken und Produkten. Sämtliche dieser Namen sind fiktiv. Ähnlichkeiten mit Namen und Adressen tatsächlicher Unternehmen oder Personen sind zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musterprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos (d. h. ohne Zahlung an IBM) kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Wird dieses Buch als Softcopy (Book) angezeigt, erscheinen keine Fotografien oder Farbabbildungen.

## Informationen zu Programmierschnittstellen

---

Die bereitgestellten Informationen zur Programmierschnittstelle sollen Sie bei der Erstellung von Anwendungssoftware für dieses Programm unterstützen.

Dieses Handbuch enthält Informationen über vorgesehene Programmierschnittstellen, die es dem Kunden ermöglichen, Programme zu schreiben, um die Services von WebSphere MQ zu erhalten.

Diese Informationen können jedoch auch Angaben über Diagnose, Bearbeitung und Optimierung enthalten. Die Informationen zu Diagnose, Bearbeitung und Optimierung sollten Ihnen bei der Fehlerbehebung für die Anwendungssoftware helfen.

**Wichtig:** Verwenden Sie diese Diagnose-, Änderungs- und Optimierungsinformationen nicht als Programmierschnittstelle, da sie Änderungen unterliegen.

## Marken

---

IBM, das IBM Logo, ibm.com, sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml). Weitere Produkt- und Servicennamen können Marken von IBM oder anderen Unternehmen sein.

Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Dieses Produkt enthält Software, die von Eclipse Project (<http://www.eclipse.org/>) entwickelt wurde.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.







Teilenummer:

(1P) P/N: